Instituto Tecnológico de Educación Superior de Monterrey

SLP, S-labeling minimization problem

Presenta:
Rocha Avila Hugo Masharelli A01633090
Vázquez Calderón Diego Armando A00226803
Calderón Barbeito André A01401035
Cazares Gastelum Juan José A00570574

Análisis y diseño de algoritmos
Profesor: Dr. Eduardo Arturo Rodriguez Tello

b) Introducción

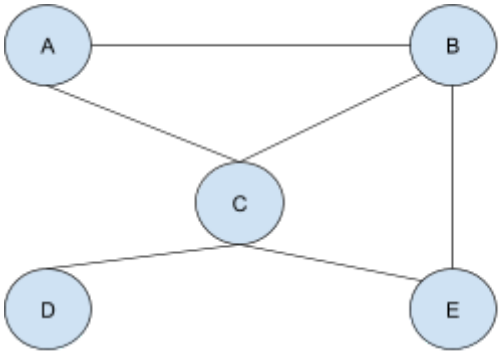
La últimas décadas se ha estudiado ampliamente los problemas de *Graph labeling* en las cuales tienen una gran área de aplicación, mencionando unos ejemplos tales como: *coding theory, computational biology, computer networks, design of error-correcting codes* etc…

En estos problemas se trata de asignar enteros positivo a los nodos o a los bordes del grafo sujeto a restricciones, tal que la función pueda ser optimizada. Para este trabajo nos basaremos en la definición siguiente del problema de *Graph labeling.*

c) Definición formal del problema:

Dado un grafo G = (V,E) , en donde G tiene todos sus vértices están conectados, de orden n y máximo grado Δ, el problema consiste en encontrar un etiquetado para G, teniendo un mapeo biyectivo con $\varphi : V = \{1, 2 \ldots n\}$, tal que $SL\varphi(G) = \{u,v\} \in E \min\{\varphi(u), \varphi(v)\}$ es mínimo.

d) Ejemplo para Ilustrar el Problema



Tenemos un grafo con 5 nodos y 6 aristas en las que se realizarán 2 permutaciones para demostrar el proceso. A cada uno de los nodos se le asigna una etiqueta numérica aleatoriamente. Por lo tanto se tienen el siguiente vector de etiquetas ordenado de acuerdo a la letra del nodo

$V = \{4, 2, 5, 1, 3\}$

A continuación, por cada una de las aristas, se toman las etiquetas de los nodos involucrados en la arista y se comparan para determinar cuál etiqueta es menor y dicha etiqueta será añadida a la sumatoria. De esta manera obtenemos la tabla siguiente

Arista	Etiquetas	Mínimo
a, b	4, 2	2
a, c	4, 5	4
b, c	2, 5	2
b, e	2, 3	2
c, d	5, 1	1
c, e	5, 3	3

La sumatoria de cada uno de los minimo da como resultado 14. Observemos que es lo que ocurre al cambiar nuestra permutación. Nuestra nueva permutación intercambiara las etiquetas de los nodos c y d. Esto nos regresa un nuevo vector y una nueva tabla

$V = \{4, 2, 1, 5, 3\}$

Arista	Etiquetas	Mínimo
a, b	4, 2	2
a, c	4, 1	1
b, c	2, 1	1
b, e	2, 3	2
c, d	1, 5	1
c, e	1, 3	1

Cómo se puede apreciar. La sumatoria resultante de esta tabla es 8, por lo que sabemos que esta nueva permutación es mejor a la anterior.

e) Algoritmo a Implementar

Problemática

Se implementará un algoritmo usando la metaheuristica ILS, primero se generarán las uniones del grafo, después se tomará una etiqueta aleatoria y su vecino y se les hace swap. Se revisa si la lista de etiquetas con el swap es mejor que la anterior, de ser mejor se guardan los valores. Después se regresa el arreglo a como estaba. Se consigue el mínimo local y las mejores etiquetas hasta el momento. Después se realizan permutaciones y se le hace una búsqueda local al arreglo permutado. Se compara la lista de etiquetas con la anterior, en caso de ser mejor, se vuelve el nuevo óptimo y se repite el proceso dependiendo del número de nodos del grafo.

Pseudocódigo:

```
minVal
etiquetas

SLP2
        minVal = V*V
        etiquetas = int[V]
endSLP2

min(a, b)
        if a<b
                return a
        elif
                return b
        endif
endmin

SLP(v, e, aritsas)
        for i=0 i<etiquetas.length i++
                etiquetas[i] = i+1
        endfor
        uniones = GenerarUniones(aristas)
        etiquetas = busquedaLocal(etiquetas,uniones,minVal)

        for i=0 i<V i++
                permutacion = permutar(etiquetas)
                permutacion = busquedaLocal(etiquetas,uniones,minVal)
```

```
                evalPermutacion = evaluar(permutacion, uniones)

                if evalPermutacion < minVal
                        minVal = evalPermutacion
                        for x=0 x<etiquetas.length x++
                                etiquetas[x] = permutacion[x]
                        endfor
                endif
        endfor
        return etiquetas
endSLP

generarUniones(aristas)
        uniones = int[aristas.length][2]
        for i = 0 i<aristas.length i++
                temp = aristas[i] dividido por espacio
                uniones[i][0] = temp[0]
                uniones[i][1] = temp[1]
        endfor
        return uniones
endgenerarUniones

busquedaLocal(etiquetas, uniones, minVal
        bestI = 0
        bestJ = 0
        minimoLocal = minVal
        orden = int[etiquetas.lenght]
        for x=0 x<etiquetas.length x++
                orden[x]=etiquetas[x]
        endfor
        for k = 0 k<100 k++
                List<int> listaorden = orden
                shuffle(listaorden)
                for i=0 i<etiquetas.length -1 i++
                        u = orden[i]
                        for j=u+1 j<etiquetas.length j++
                                temp = etiquetas[u]
                                etiquetas[u] = etiquetas[j]
                                etiquetas[j] = temp
                                suma = evaluar(etiquetas, uniones)
                                if minimolocal>suma
                                        minimolocal = suma
```

```
                                        bestI = u
                                        nestJ = j
                                endif
                                temp = etiquetas[u]
                                etiquetas[u] = etiquetas[j]
                                etiquetas[j] = temp
                        endfor
                endfor
        endfor
        temp = etiquetas[bestI]
        etiquetas[bestI] = etiquetas[bestJ]
        etiquetas[bestJ] = temp
        minimoLocal = evaluar(etiquetas,uniones)
        return etiquetas
endbusquedaLocal

evaluar(etiquetas,uniones)
        sum = 0
        for k=0 k<uniones.length k++
                a = etiquetas[uniones[k][0]-1]
                b = etiquetas[uniones[k][1]-1]
                sum= sum + min(a,b)
        endfor
        return sum
endevaluar

permutarEtiquetas(arr)
        temp = arr[0]
        for i=0 i<arr.length-1 i++
                arr[i] = arr[i+1]
        endfor
        arr[arr.length-1]  = temp
        return arr
endpermutarEtiquetas
```

Análisis Matemático:

$$\sum_{n=0}^{etiquetas.length} 1 + \sum_{n=0}^{aristas.length} 1 + \sum_{n=0}^{etiquetas.length} 1 + \sum_{n=0}^{100} \left(\sum_{j=0}^{etiquetas.length} \left(\sum_{x=y}^{etiquetas.length} 2 \right) \right) + \sum_{n=0}^{\infty} \sum_{x=0}^{etiquetas.length-1} 1 + \sum_{n=0}^{100} \sum_{j=0}^{etiquetas.length} \left(\sum_{x=y}^{etiquetas.length} 2 \right) + 1 + \sum_{l=0}^{etiquetas.length} \sum_{x=0}^{etiquetas.length} 1$$

g) Experimentación con Algoritmo:

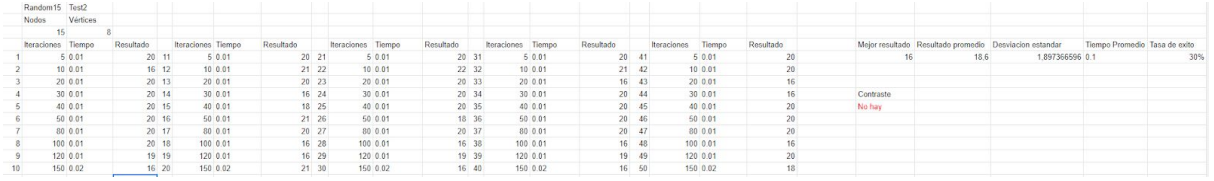
Decidimos que los experimentos serían con 15 diferentes instancias las cuales van a tener 10 iteraciones diferentes con las cuales calculamos 2 resultados tiempo de ejecución y el resultado.

Repitiendo esto 50 veces para tener una mejor precisión, enseguida se anexan imágenes de los resultados que igualmente se entregarán al final.

Nodos: 10 Aristas:4

Random10	Test1																				
Nodos	Aristas																				
10	4																				
Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
1	5	0.01	8	11	5	0.01	8	21	5	0.01	8	31	5	0.01	8	41	5	0.01	8	8	0	0.01	100%
2	10	0.01	8	12	10	0.01	8	22	10	0.01	8	32	10	0.01	8	42	10	0.01	8				
3	20	0.01	8	13	20	0.01	8	23	20	0.01	8	33	20	0.01	8	43	20	0.01	8				
4	30	0.01	8	14	30	0.01	8	24	30	0.01	8	34	30	0.01	8	44	30	0.01	8	Contraste			
5	40	0.01	8	15	40	0.01	8	25	40	0.01	8	35	40	0.01	8	45	40	0.01	8	No hay			
6	50	0.01	8	16	50	0.01	8	26	50	0.01	8	36	50	0.01	8	46	50	0.01	8				
7	80	0.01	8	17	80	0.01	8	27	80	0.01	8	37	80	0.01	8	47	80	0.01	8				
8	100	0.01	8	18	100	0.01	8	28	100	0.01	8	38	100	0.01	8	48	100	0.01	8				
9	120	0.01	8	19	120	0.01	8	29	120	0.01	8	39	120	0.01	8	49	120	0.01	8				
10	150	0.02	8	20	150	0.02	8	30	150	0.02	8	40	150	0.01	8	50	150	0.02	8				

Nodos : 15 Vértices: 8



Random15	Test2																				
Nodos	Vértices																				
15	8																				
Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
1	5	0.01	20	11	5	0.01	20	21	5	0.01	20	31	5	0.01	20	41	5	0.01	20	16	18.6	1.897366596	0.1	30%
2	10	0.01	16	12	10	0.01	21	22	10	0.01	22	32	10	0.01	21	42	10	0.01	20					
3	20	0.01	20	13	20	0.01	20	23	20	0.01	20	33	20	0.01	16	43	20	0.01	16					
4	30	0.01	20	14	30	0.01	16	24	30	0.01	20	34	30	0.01	20	44	30	0.01	16	Contraste				
5	40	0.01	20	15	40	0.01	18	25	40	0.01	20	35	40	0.01	20	45	40	0.01	20	No hay				
6	50	0.01	20	16	50	0.01	21	26	50	0.01	18	36	50	0.01	20	46	50	0.01	20					
7	80	0.01	20	17	80	0.01	20	27	80	0.01	20	37	80	0.01	20	47	80	0.01	20					
8	100	0.01	20	18	100	0.01	16	28	100	0.01	16	38	100	0.01	16	48	100	0.01	16					
9	120	0.01	19	19	120	0.01	16	29	120	0.01	19	39	120	0.01	19	49	120	0.01	20					
10	150	0.02	16	20	150	0.02	21	30	150	0.02	16	40	150	0.02	16	50	150	0.02	18					

Nodos :20 Vértices: 29

Random20	Test3																				
Nodos	Vértices																				
20	29																				
Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado		Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
1	5	0.012	141	11	5	0.012	141	21	5	0.012	141	31	5	0.012	141	41	5	0.012	141	141	141	0	0.028	100%
2	10	0.016	141	12	10	0.016	141	22	10	0.016	141	32	10	0.018	141	42	10	0.017	141					
3	20	0.019	141	13	20	0.019	141	23	20	0.018	141	33	20	0.018	141	43	20	0.019	141					
4	30	0.027	141	14	30	0.026	141	24	30	0.027	141	34	30	0.026	141	44	30	0.027	141	Contraste				
5	40	0.028	141	15	40	0.027	141	25	40	0.027	141	35	40	0.027	141	45	40	0.028	141	No hay				
6	50	0.029	141	16	50	0.028	141	26	50	0.029	141	36	50	0.028	141	46	50	0.029	141					
7	80	0.038	141	17	80	0.038	141	27	80	0.039	141	37	80	0.038	141	47	80	0.039	141					
8	100	0.068	141	18	100	0.057	141	28	100	0.057	141	38	100	0.068	141	48	100	0.057	141					
9	120	0.068	141	19	120	0.057	141	29	120	0.068	141	39	120	0.057	141	49	120	0.068	141					
10	150	0.063	141	20	150	0.062	141	30	150	0.063	141	40	150	0.062	141	50	150	0.062	141					

Nodos:25 Vértices:48

Random25	Test4																				
Nodos	Vértices																				
25	48																				
Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Iteraciones	Tiempo	Resultado	Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
1	5	0.018	276	11	5	0.022	276	21	5	0.022	276	31	5	0.022	276	41	5	0.025	276	276	277.5	3.566822427	0.074	80%
2	10	0.025	276	12	10	0.026	276	22	10	0.025	276	32	10	0.026	276	42	10	0.025	276					
3	20	0.046	276	13	20	0.033	276	23	20	0.044	276	33	20	0.033	276	43	20	038	276					
4	30	0.044	276	14	30	0.044	276	24	30	0.048	276	34	30	0.044	276	44	30	0.047	276	Contraste				
5	40	0.055	276	15	40	0.053	276	25	40	0.054	276	35	40	0.056	276	45	40	0.058	276	No hay				
6	50	0.071	276	16	50	0.075	276	26	50	0.074	276	36	50	0.071	276	46	50	0.074	276					
7	80	0.102	276	17	80	0.091	276	27	80	0.092	276	37	80	0.093	276	47	80	0.093	276					
8	100	0.119	276	18	100	0.107	276	28	100	0.143	276	38	100	0.095	276	48	100	0.095	276					
9	120	0.132	287	19	120	0.142	287	29	120	0.14	287	39	120	0.128	287	49	120	0.128	287					
10	150	0.129	280	20	150	0.123	276	30	150	0.124	280	40	150	0.123	280	50	150	0.124	280					

Nodos:35 Vértices:113

Random35	Test5
Nodos	Vértices
35	113

	Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado
1	5	0.058	1087	11	5	0.058	1102	21	5	0.058	1102	31	5	0.058	1102	41	5	0.058	1102
2	10	0.106	1065	12	10	0.092	1089	22	10	0.089	1065	32	10	0.089	1163	42	10	0.089	1065
3	20	0.185	1088	13	20	0.173	1088	23	20	0.185	1088	33	20	0.173	1088	43	20	0.106	1088
4	30	0.244	1088	14	30	0.258	1088	24	30	0.244	1088	34	30	0.258	1088	44	30	0.244	1088
5	40	0.39	1088	15	40	0.393	1088	25	40	0.393	1088	35	40	0.39	1088	45	40	0.393	1088
6	50	0.438	1088	16	50	0.475	1088	26	50	0.438	1088	36	50	0.475	1088	46	50	0.438	1088
7	80	0.751	1088	17	80	0.721	1088	27	80	0.751	1088	37	80	0.649	1088	47	80	0.684	1088
8	100	0.848	1088	18	100	0.848	1088	28	100	0.879	1088	38	100	0.879	1088	48	100	0.782	1088
9	120	0.949	1082	19	120	1.117	1082	29	120	1.036	1082	39	120	1.036	1082	49	120	0.949	1082
10	150	1.166	1088	20	150	1.117	1088	30	150	1.166	1088	40	1.212	1.166	1088	50	150	1.166	1088

Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
1065	1086,5	9,058452161	0.438	10%

Contraste
No hay

Nodos:40 Vertices:165

Nodos	Vértices
40	165

	Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado
1	5	0.098	1600	11	5	0.103	1600	21	5	0.112	1600	31	5	0.103	1600	41	5	0.098	1600
2	10	0.188	1600	12	10	0.198	1600	22	10	0.188	1600	32	10	0.198	1600	42	10	0.204	1600
3	20	0.351	1600	13	20	0.38	1600	23	20	0.386	1600	33	20	0.379	1600	43	20	0.386	1600
4	30	0.559	1600	14	30	0.559	1600	24	30	0.524	1600	34	30	0.559	1600	44	30	0.524	1600
5	40	0.542	1600	15	40	0.571	1600	25	40	0.571	1600	35	40	0.582	1600	45	40	0.542	1600
6	50	0.917	1600	16	50	0.894	1600	26	50	0.925	1600	36	50	0.825	1600	46	50	0.925	1600
7	80	1.322	1600	17	80	1.283	1600	27	80	1.342	1600	37	80	1.435	1600	47	80	1.342	1600
8	100	1.743	1600	18	100	1.684	1600	28	100	1.711	1600	38	100	1.684	1600	48	100	1.711	1600
9	120	2.133	1600	19	120	2.118	1600	29	120	2.017	1600	39	120	2.118	1600	49	120	2.017	1600
10	150	2.46	1600	20	150	2.567	1600	30	150	2.437	1600	40	150	2.567	1600	50	150	2.62	1600

Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
1600	1600	0	**2.017,00**	100%

Contraste
No hay

Nodos:45 Vértices:206

Random45	Test7
Nodos	Aristas
45	206

	Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado
1	5	0,107	2025	11	5	0,146	2025	21	5	0,107	2025	31	5	0,113	2025	41	5	0,118	2025
2	10	0,208	2025	12	10	0,229	2025	22	10	0,188	2025	32	10	0,395	2025	42	10	0,203	2025
3	20	0,42	2025	13	20	0,381	2025	23	20	0,367	2025	33	20	0,359	2025	43	20	0,415	2025
4	30	0,598	2025	14	30	0,51	2025	24	30	0,678	2025	34	30	0,555	2025	44	30	0,559	2025
5	40	0,808	2025	15	40	0,737	2025	25	40	0,687	2025	35	40	0,7	2025	45	40	0,748	2025
6	50	1	2025	16	50	0,893	2025	26	50	0,896	2025	36	50	0,964	2025	46	50	0,966	2025
7	80	1,394	2025	17	80	1,387	2025	27	80	1,521	2025	37	80	1,434	2025	47	80	1,492	2025
8	100	1,954	2025	18	100	1,736	2025	28	100	1,689	2025	38	100	1,829	2025	48	100	1,875	2025
9	120	2,307	2025	19	120	1,973	2025	29	120	2,059	2025	39	120	3,817	2025	49	120	2,298	2025
10	150	2,664	2025	20	150	3,041	2025	30	150	2,842	2025	40	150	5,27	2025	50	150	2,927	2025

Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
2025	2025	0	1,1601	100%

Contraste
No hay

Nodos:50 Vértices: 275

Nodos	Aristas
50	275

	Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo			Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado
1	5	0,23	2500	11	5	0,183	2500	21	5	0,217	2500	31	5	0,236	2500	41	5	0,19	2500
2	10	0,514	2500	12	10	0,38	2500	22	10	0,366	2500	32	10	0,347	2500	42	10	0,338	2500
3	20	0,716	2500	13	20	0,839	2500	23	20	0,697	2500	33	20	0,797	2500	43	20	0,688	2500
4	30	1,045	2500	14	30	0,963	2500	24	30	1,103	2500	34	30	0,964	2500	44	30	0,984	2500
5	40	1,394	2500	15	40	1.301	2500	25	40	1,405	2500	35	40	1,295	2500	45	40	1,342	2500
6	50	1,606	2500	16	50	1,655	2500	26	50	1,716	2500	36	50	1,666	2500	46	50	1,627	2500
7	80	2,854	2500	17	80	2,584	2500	27	80	2,585	2500	37	80	2,821	2500	47	80	2,636	2500
8	100	3,44	2500	18	100	3,436	2500	28	100	3,31	2500	38	100	3,132	2500	48	100	3,22	2500
9	120	4,879	2500	19	120	5,147	2500	29	120	4,03	2500	39	120	5,032	2500	49	120	3,934	2500
10	150	5,091	2500	20	150	5,098	2500	30	150	5,022	2500	40	150	5,02	2500	50	150	5,001	2500

Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
2500	2500	0	1,996	100%

Contraste
No hay

Nodos:60 Vértices:435

Random60	Test9
Nodos	Aristas
60	435

	Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado
1	5	0,431	3600	11	5	0,434	3600	21	5	0,465	3600	31	5	0,511	3600	41	5	0,459	3600
2	10	0,917	3600	12	10	0,838	3600	22	10	0,934	3600	32	10	0,965	3600	42	10	1,045	3600
3	20	1,708	3600	13	20	1,695	3600	23	20	1,77	3600	33	20	1,983	3600	43	20	1,842	3600
4	30	2,558	3600	14	30	2,557	3600	24	30	2,621	3600	34	30	2,705	3600	44	30	2,742	3600
5	40	3,747	3600	15	40	3,463	3600	25	40	3,587	3600	35	40	3,745	3600	45	40	6,396	3600
6	50	6,767	3600	16	50	4,799	3600	26	50	4,559	3600	36	50	4,543	3600	46	50	5,117	3600
7	80	14,239	3600	17	80	8,383	3600	27	80	8,037	3600	37	80	7,247	3600	47	80	7,831	3600
8	100	9,888	3600	18	100	9,872	3600	28	100	9,758	3600	38	100	10,626	3600	48	100	10,461	3600
9	120	11,323	3600	19	120	12,192	3600	29	120	11,556	3600	39	120	12,336	3600	49	120	12,475	3600
10	150	14,766	3600	20	150	15,089	3600	30	150	15,153	3600	40	150	20,3	3600	50	150	15,627	3600

Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
2025	2025	0	6,3995	100%

Contraste
No hay

Nodos:70 Vértices:565

Random70	Test10
Nodos	Aristas
70	565

	Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado
1	5	0,99	4900	11	5	1,083	4900	21	5	0,987	4900	31	5	1,013	4900	41	5	1,02	4900
2	10	2,055	4900	12	10	1,925	4900	22	10	2,035	4900	32	10	2,124	4900	42	10	2,014	4900
3	20	3,94	4900	13	20	4,942	4900	23	20	5,446	4900	33	20	4,001	4900	43	20	3,909	4900
4	30	6,235	4900	14	30	9,759	4900	24	30	5,995	4900	34	30	5,803	4900	44	30	6,232	4900
5	40	8,366	4900	15	40	7,931	4900	25	40	8,406	4900	35	40	7,915	4900	45	40	7,638	4900
6	50	10,534	4900	16	50	14,763	4900	26	50	14,623	4900	36	50	10,221	4900	46	50	10,324	4900
7	80	16,809	4900	17	80	16,117	4900	27	80	30,536	4900	37	80	15,896	4900	47	80	16,462	4900
8	100	20,243	4900	18	100	20,161	4900	28	100	21,9	4900	38	100	20,27	4900	48	100	20,003	4900
9	120	24,752	4900	19	120	24,832	4900	29	120	26,69	4900	39	120	23,906	4900	49	120	25,857	4900
10	150	30,122	4900	20	150	30,754	4900	30	150	32,218	4900	40	150	31,207	4900	50	150	33,132	4900

Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
2025	2025	0	12,6591	100%

Contraste
No hay

Nodos:80 Vértices:721

Random80	Test11
Nodos	Vértices
80	721

	Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado
1	5	2.385	6400	11	5	2.311	6400	21	5	02.04	6400	31	5	2.22	6400	41	5	2.354	6400
2	10	4.81	6400	12	10	4.70	6400	22	10	4.99	6400	32	10	4.83	6400	42	10	4.98	6400
3	20	8.069	6400	13	20	8.112	6400	23	20	8.74	6400	33	20	8.131	6400	43	20	8.099	6400
4	30	12.56	6400	14	30	12.12	6400	24	30	12.78	6400	34	30	12.99	6400	44	30	12.341	6400
5	40	15.64	6400	15	40	15.77	6400	25	40	15.93	6400	35	40	15.63	6400	45	40	15.28	6400
6	50	18.75	6400	16	50	18.24	6400	26	50	18.93	6400	36	50	18.26	6400	46	50	18.53	6400
7	80	32.02	6400	17	80	31.41	6400	27	80	31.88	6400	37	80	32.71	6400	47	80	31.151	6400
8	100	41.87	6400	18	100	42.02	6400	28	100	41.56	6400	38	100	41.36	6400	48	100	42.001	6400
9	120	49.43	6400	19	120	50.12	6400	29	120	48.3	6400	39	120	52.11	6400	49	120	49.33	6400
10	150	65.378	6400	20	150	61.389	6400	30	150	63.44	6400	40	150	61.587	6400	50	150	62.87	6400

Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
2025	2025	0	19.189	100%

Contraste
No hay

Nodos:90 Vértices:930

	Random90	Test12																	
	Nodos	Vértices																	
	90	930																	
	Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado
1	5	3.51	8100	11	5	3.42	8100	21	5	3.27	8100	31	5	3.54	8100	41	5	3.24	8100
2	10	7.24	8100	12	10	7.46	8100	22	10	7.35	8100	32	10	7.59	8100	42	10	7.37	8100
3	20	11.43	8100	13	20	11.36	8100	23	20	11.87	8100	33	20	11.21	8100	43	20	11.61	8100
4	30	21.45	8100	14	30	22.01	8100	24	30	21.82	8100	34	30	21.90	8100	44	30	21.39	8100
5	40	28.40	8100	15	40	28.11	8100	25	40	27.92	8100	35	40	28.03	8100	45	40	28.4	8100
6	50	35.41	8100	16	50	35.02	8100	26	50	35.73	8100	36	50	35.75	8100	46	50	27.87	8100
7	80	56.72	8100	17	80	56.90	8100	27	80	56.32	8100	37	80	56.95	8100	47	80	56.21	8100
8	100	73.36	8100	18	100	72.87	8100	28	100	73.14	8100	38	100	73.00	8100	48	100	73.81	8100
9	120	95.984	8100	19	120	94.98	8100	29	120	93.87	8100	39	120	95.28	8100	49	120	95.55	8100
10	150	114.347	8100	20	150	112.46	8100	30	150	114..65	8100	40	150	115.11	8100	50	150	114.91	8100

Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
2025	2025	0	43949	100%
Contraste				
No hay				

Nodos:100 Vértices:1220

	Random100	Test13																	
	Nodos	Vértices																	
	100	1220																	
	Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado
1	5	8.475	10000	11	5	8.475		21	5	8.475	10000	31	5	8.475	10000	41	5	8.475	10000
2	10	17.341	10000	12	10	17.341		22	10	17.341	10000	32	10	17.341	10000	42	10	17.341	10000
3	20	32.118	10000	13	20	32.154		23	20	31.125	10000	33	20	32.175	10000	43	20	32.153	10000
4	30	49.79	10000	14	30	49.34		24	30	49.54	10000	34	30	50.123	10000	44	30	51.532	10000
5	40	66.51	10000	15	40	66.43		25	40	66.54	10000	35	40	66.123	10000	45	40	66.64	10000
6	50	78.23	10000	16	50	78.53		26	50	78.543	10000	36	50	79.43	10000	46	50	78.34	10000
7	80	98.43	10000	17	80	99.12		27	80	99.12	10000	37	80	98.35	10000	47	80	98.64	10000
8	100	114.347	10000	18	100	112.46		28	100	115.11	10000	38	100	114.91	10000	48	100	114..65	10000
9	120	235.967	10000	19	120	243.521		29	120	243.543	10000	39	120	246.523	10000	49	120	243.643	10000
10	150	255.967	10000	20	150	255.967		30	150	255.967	10000	40	150	255.967	10000	50	150	255.967	10000

Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
2025	2025	0	#¡DIV/0!	100%
Contraste				
No hay				

Nodos:120 Vértices:1684

	Random120	Test14																	
	Nodos	Vértices																	
	120	1684																	
	Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado
1	5	30.543	14400	11	5	30.965	14400	21	5	30.635	14400	31	5	30.735	14400	41	5	30.965	14400
2	10	43.369	14400	12	10	43.723	14400	22	10	43.593	14400	32	10	43.467	14400	42	10	43.896	14400
3	20	86.46	14400	13	20	86.54	14400	23	20	86.25	14400	33	20	86.75	14400	43	20	86.63	14400
4	30	100.35	14400	14	30	100.524	14400	24	30	100.632	14400	34	30	100.623	14400	44	30	100.847	14400
5	40	102.412	14400	15	40	102.15	14400	25	40	102.62	14400	35	40	102.89	14400	45	40	102.789	14400
6	50	103.523	14400	16	50	103.523	14400	26	50	103.523	14400	36	50	103.523	14400	46	50	103.523	14400
7	80	254.512	14400	17	80	254.643	14400	27	80	254.763	14400	37	80	254.843	14400	47	80	254.512	14400
8	100	486.432	14400	18	100	486.624	14400	28	100	486.745	14400	38	100	486.845	14400	48	100	486.753	14400
9	120	509.523	14400	19	120	509.645	14400	29	120	509.734	14400	39	120	509.623	14400	49	120	509.634	14400
10	150	607.424	14400	20	150	607.523	14400	30	150	607.643	14400	40	150	607.634	14400	50	150	607.654	14400

Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
2025	2025	0	103.523	100%
Contraste				
No hay				

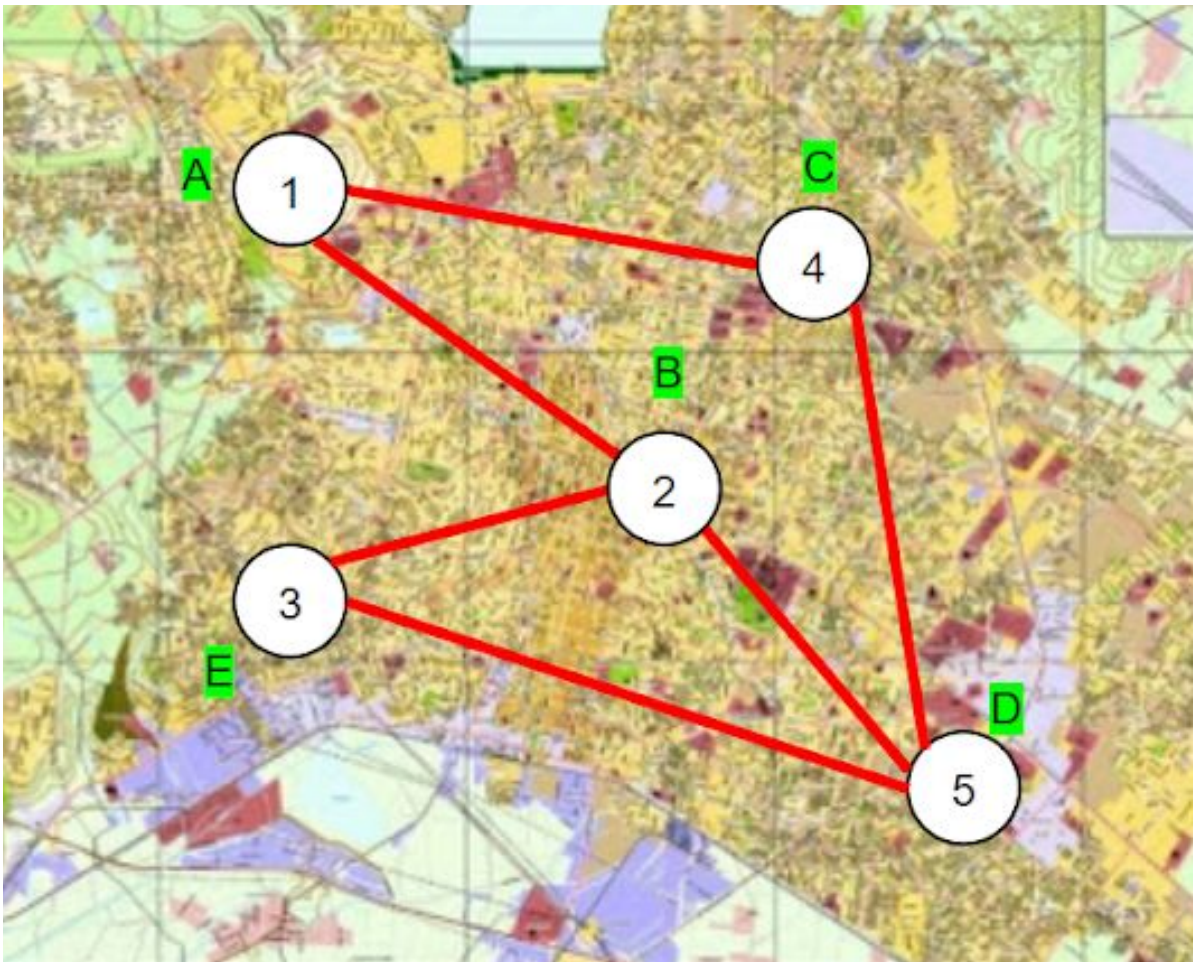
Nodos:140 Vertices:

	Random140	Test15																	
	Nodos	Vértices																	
	140																		
	Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado		Iteraciones	Tiempo	Resultado
1	5	48.555	14400	11	5	48.723	14400	21	5	48.663	14400	31	5	48.734	14400	41	5	48.834	14400
2	10	86.65	14400	12	10	86.745	14400	22	10	86.734	14400	32	10	86.235	14400	42	10	86.754	14400
3	20	97.765	14400	13	20	97.6345	14400	23	20	97.724	14400	33	20	97.872	14400	43	20	97.964	14400
4	30	100.523	14400	14	30	100.523	14400	24	30	100.523	14400	34	30	100.523	14400	44	30	100.523	14400
5	40	108.734	14400	15	40	108.734	14400	25	40	108.734	14400	35	40	108.734	14400	45	40	108.734	14400
6	50	106.575	14400	16	50	106.575	14400	26	50	106.575	14400	36	50	106.575	14400	46	50	106.575	14400
7	80	264.234	14400	17	80	264.234	14400	27	80	264.234	14400	37	80	264.234	14400	47	80	264.234	14400
8	100	466.489	14400	18	100	466.489	14400	28	100	466.489	14400	38	100	466.489	14400	48	100	466.489	14400
9	120	579.562	14400	19	120	579.562	14400	29	120	579.562	14400	39	120	579.562	14400	49	120	579.562	14400
10	150	601.941	14400	20	150	601.895	14400	30	150	601.934	14400	40	150	601.885	14400	50	150	601.843	14400

Mejor resultado	Resultado promedio	Desviacion estandar	Tiempo Promedio	Tasa de exito
2025	2025	0	106.575	100%
Contraste				
No hay				

h) Situación práctica de la vida real:

Supongamos que se tiene el plano de una ciudad en donde se quiere crear un sistema de metro y hasta el momento se tienen las estaciones que serán construidas y las aristas son las vías de metro que conectará las estaciones.



Cada nodo tiene un etiquetado que representa el número durante la búsqueda local, lo que se quiere lograr con este problema es obtener la mínima suma de todas las conexiones entre nodos. Por ejemplo, en la imagen de arriba tenemos las conexiones:
A - C, A - B, B - E, B - D, C - D, E - D.

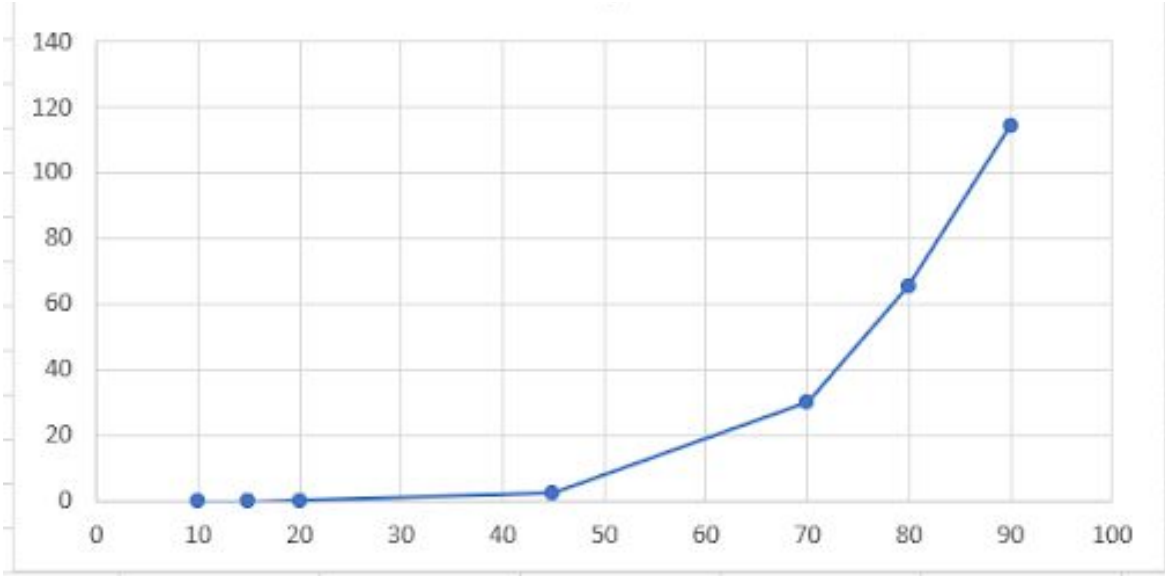
Entonces, a la hora de obtener los mínimos en este caso serían:
El mínimo entre 1 y 4 es 1;
El mínimo entre 1 y 2 es 1;
El mínimo entre 2 y 3 es 2;
El mínimo entre 2 y 5 es 2;
El mínimo entre 4 y 5 es 4;
El mínimo entre 3 y 5 es 3;

A la hora de hacer la suma de los mínimos nos da como resultado 13, entonces lo que se busca es realizar permutaciones entre etiquetas para lograr obtener el mínimo resultado posible.

En este ejemplo práctico la solución serviría para asignar relevancia a las estaciones de metro en la ciudad y así poder construir estaciones adecuadas con base en su concurrencia.

Dado que la solución óptima para esta clase de problema requeriría hacer todas las permutaciones posibles entre etiquetas y esto tomaría demasiado tiempo de ejecución para grafos muy grandes. Nuestra solución fue permutar las etiquetas de manera aleatoria dado cierto número de iteraciones y así tratar el mejor resultado posible o un aproximado para todos los casos posibles.

Con base en nuestro análisis matemático acerca de nuestro algoritmo, pudimos obtener la siguiente gráfica de convergencia:



En donde el eje X es el tamaño del grafo aleatoriamente conectado y el eje Y es el tiempo que tarda en ejecutar las iteraciones ingresadas.

Ingresando esta instancia del problema con 150 iteraciones en nuestro algoritmo obtuvimos los siguientes resultados:

```
El valor mínimo aproximado es: 10

La mejor etiqueta para 1 es 1
La mejor etiqueta para 2 es 2
La mejor etiqueta para 3 es 3
La mejor etiqueta para 4 es 5
La mejor etiqueta para 5 es 2
Current Time in milliseconds = 0.002
```

Todas las permutaciones posibles para esta instancia habría tomado 3,125 iteraciones, que sería muy costoso en términos de ejecución. Por lo que concluímos que nuestro resultado es una muy buena aproximación para todas las instancias de nuestro problema.

i) Aportaciones:

Juan José	Mi labor como miembro del equipo fue ayudar a crear el algoritmo que pudiera resolver nuestro problema. Otra labor fue diseñar las instancias de prueba, que pudieran de cierto modo poner a prueba los extremos de nuestro algoritmo. También me dedique a realizar el análisis de las pruebas resultantes. Realicé el ejemplo práctico en la vida real.
Hugo	Mi contribución durante el proyecto fue la investigación de algoritmos y formas de poder resolver el problema de SLP, por lo que después ayude a la creación del algoritmo. Después de ello tuve participación en las correcciones y mejora del código, al igual que las pruebas que se hicieron en este. También a sacar la estadística y datos de los experimentos que se realizaron. Al final a la recopilación de la información y formato para la entrega.
Diego	Durante la realización del proyecto, mis aportaciones se enfocaron principalmente a mejorar el código de nuestro programa para poder implementar el algoritmo metaheurístico de la mejor manera. Al igual que mis compañeros, realice varias pruebas de las instancias que se generaron.
André	Participe en la investigación de algoritmos metaheurísticos y heurísticos para logra diferenciarlos y poder tener una perspectivas más eficiente, al igual que la realización de pseudocodigo para el algoritmo de nuestra solución. A

	la par se realizó el análisis matemático de cada uno de los algoritmos que se fueron desarrollando durante la realización del proyecto.Al igual se ayudo en la realización de los experimentos con la finalidad de probar el algoritmo final y obtener los datos para las gráficas y estadísticas.

j) Aprendizajes:

	Lo que aprendimos	Lo que nos gustó del curso	Lo que no nos gustó del curso
Juan José	Durante el semestre logré aprender lo que esperaba de la materia de algoritmos y un poco más. Los trabajos y tareas fueron retadores pero logré comprender todos los temas.	Lo que más me agradó del curso fue lo retadores que son los algoritmos y las posibilidades que ofrecen a la solución de problemas de la vida cotidiana.	No me gustó que no hubo mucho trabajo práctico en cuanto a programación de algoritmos.
Hugo	Que existe una infinidad de temas acerca de los algoritmos que desconocemos, al igual que tenemos diferentes tipos de aproximaciones o soluciones hacia estos y como podemos implementarlos en la vida diaria, cosa que nos ayudará en nuestro crecimiento	La variación de temas que se tocaron durante el curso fue lo mejor al igual que el enfoque que se le dio a los problemas ya que en experiencia propia me ayudaron demasiado en entrevistas de trabajo.	La falta de ejercicios en cuestión a un poco más específicos, al igual enfocarnos un poco más en ciertos tópicos que pueden llegar a ser más importantes.

	profesional y personal.		
Diego	Durante el curso, pude percatarme de la importancia de los algoritmos para solución de problemas. Pude ver que un algoritmo específico puede ser útil en diversos escenarios y que es muy importante el análisis del mismo ya que esto puede llevar a un mayor entendimiento del algoritmo y mejoras en el futuro	Me gustó que a lo largo del semestre pudimos ver una variedad de temas muy interesantes que nos pueden ser útiles más adelante. Por ejemplo, el análisis de la complejidad computacional llamó mi atención	En algunas ocasiones, por la falta de tiempo, algunos temas se quedaban cortos en cuanto a la profundidad de la explicación, y a veces solo podíamos verlos desde un punto de vista teórico. Creo que nos hizo falta ver más ejemplos o realizar más actividades
André	Aprendí las habilidades necesarias para poder identificar y analizar correctamente cada uno de los algoritmos presentados y al igual poder desarrollar el análisis crítico para los futuros problemas relacionados con los temas impartidos en clase.	Me gusto demasiado el énfasis en el análisis de matemáticos de los problemas y algoritmos que me ayudaran en el futuro.	No me llego a atraer demasiado ciertos tópicos que se tocaron en la clase.

k) Conclusiones:

El proyecto final constituye un reto importante para nosotros. Sin embargo, nos brindó diversos aprendizajes. En primer lugar, pudimos percatarnos de lo complejos que pueden llegar a ser algunos problemas. Ya no basta con solamente intentar crear una solución con una heurística para obtener un resultado óptimo. Se necesita de algo diferente. Logramos aprender sobre el rol que juegan los algoritmos metaheurísticos en la generación de soluciones a problemas complejos. Además, nos dimos cuenta de cómo las implementaciones de estos algoritmos utilizan proceso no deterministas para poder llegar a soluciones considerablemente buenas. Consideramos que son una estrategia muy útil al momento de enfrentar un problema cuya solución requiere una implementación muy difícil de lograr. Definitivamente, logramos expandir nuestro conocimiento de problemas y algoritmos, lo cual nos servirá en un futuro.

l) Referencias:

Blum, C. et Roli, A. (2003) Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. Recuperado de https://drive.google.com/drive/u/1/folders/15b9CKLkwq7FejELpLrevM-4HdUEVuxxp

Sinnl, M. (2019) Algorithmic expedients for the S-labeling problem. Recuperado de https://drive.google.com/drive/u/1/folders/1EZ6Clv_a-qHqb0cR3EYYtj7YA3TxveJi

Fertin, G., Rusu, I. et Vialette, S. (2015) Algorithmic Aspects of the S-Labeling Problem. Recuperado de https://drive.google.com/drive/u/1/folders/1EZ6Clv_a-qHqb0cR3EYYtj7YA3TxveJi