



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FISICOMATEMÁTICAS



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA

PIA

Algoritmos no informado, informado y heurístico
para encontrar la ruta más corta

INTELIGENCIA ARTIFICIAL

Docente: Juan Pablo Rosas Baldazo

Grupo: 32

Alumno: Luis Javier Olivares Lara

Matricula: 1803042

14 de mayo del 2022

Contenido

SECCION 1. PLANTEAMIENTO DEL PROBLEMA.

DESCRIPCIÓN DEL PROBLEMA

DESCRIPCIÓN DEL ESTADO INICIAL Y DEL ESTADO FINAL

POSIBLES ACCIONES QUE SE PUEDEN HACER PARA RESOLVER EL PROBLEMA

AMBIENTE DEL PROBLEMA

FUNCIÓN DE COSTO

APLICACIÓN EN LA VIDA REAL

SECCION 2. DESCRICION DE METODOS DE SOLUCION

BUSQUEDA POR ANCHURA (BPA)

BÚSQUEDA VORAZ PRIMERO EL MEJOR (BVPM)

RECOCIDO SIMULADO (RS)

SECCION 3. EXPERIMENTACION

DESCRIPCIÓN DE LAS INSTANCIAS

COMO SE CONSTRUYE UNA INSTANCIA

INSTANCIAS USADAS PARA ESTA EXPERIMENTACIÓN

QUE SE ESPERA DEL EXPERIMENTO

HIPÓTESIS

SECCION 4. TABLAS Y GRAFICAS

PROMEDIOS DEL ALGORITMO BÚSQUEDA VORAZ PRIMERO EL MEJOR (BVPM)

PROMEDIOS DEL ALGORITMO BÚSQUEDA PRIMERO EN AMPLITUD (BPA)

PROMEDIOS DEL ALGORITMO RECOCIDO SIMULADO (RS)

INSTANCIAS

SECCION 5. RESULTADOS Y CONCLUSIONES

RESULTADOS ALGORITMO BVPM

RESULTADOS ALGORITMO BPA

RESULTADOS ALGORITMO RS

CONCLUSIONES SOBRE LAS INSTANCIAS

RESULTADOS ENTRE ALGORITMOS

RESULTADOS DE LOS TIPOS DE INSTANCIA DE SOLUCION

CONCLUSIONES SOBRE LOS ALGORITMOS

JUSTIFICACIÓN SOBRE LOS RESULTADOS DEL ALGORITMO RS

CODIGO

SECCION 1. PLANTEAMIENTO DEL PROBLEMA.

DESCRIPCIÓN DEL PROBLEMA

El problema es encontrar la ruta más corta en un grafo.

DESCRIPCIÓN DEL ESTADO INICIAL Y DEL ESTADO FINAL

El estado inicial y final es representado por la imagen 1 y 2, el nodo inicial es el nodo 2 y el nodo final es el nodo 5.

ESTADO INICIAL

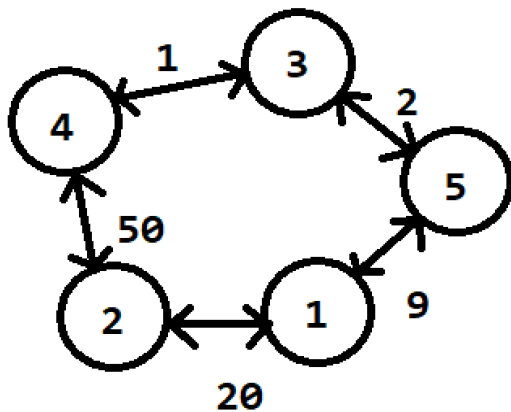


Imagen 1

ESTADO FINAL

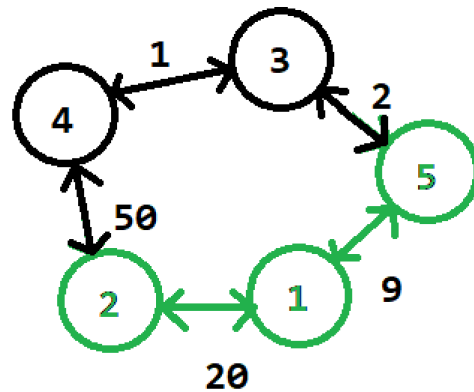


Imagen 2

Como entrada a todos los algoritmos usan un diccionario, el grafo generado es una lista de adyacencia que se guarda en un documento de texto que posteriormente se introduce a un diccionario.

```
{'1': ['5', '2'],  
 '2': ['4', '1', '3', '5'],  
 '3': ['2', '5', '4'],  
 '4': ['3', '2'],  
 '5': ['2', '1', '3'],  
 'Distancias.1': [5, 17],  
 'Distancias.2': [25, 17, 22, 20],  
 'Distancias.3': [22, 21, 3],  
 'Distancias.4': [3, 25],  
 'Distancias.5': [20, 5, 21]}
```

Imagen 3. Un grafo con 5 nodos en un diccionario.

```
PIA_IA > inst_Grafos.txt
1  ['5', '2']
2  [5, 17]
3  ['4', '1', '3', '5']
4  [25, 17, 22, 20]
5  ['2', '5', '4']
6  [22, 21, 3]
7  ['3', '2']
8  [3, 25]
9  ['2', '1', '3']
10 [20, 5, 21]
11
```

Imagen 4. El grafo se guarda en un documento de texto a manera de listas de adyacencia, para después ser leído e insertado en un diccionario.

Como resultado, los algoritmos regresan una lista ordenada con los nodos a recorrer, con la distancia total, y para propósitos del PIA nos regresa el tiempo de cómputo.

POSIBLES ACCIONES QUE SE PUEDEN HACER PARA RESOLVER EL PROBLEMA

Permitidas

- Poder moverte de nodo a un nodo adyacente.

Prohibidas

- Moverse a un nodo que no es adyacente al nodo actual (teletransportarse)
- Visitar un nodo ya recorrido
- Hacer recorridos en bucle

AMBIENTE DEL PROBLEMA

El grafo es conexo, no dirigido y con costos, el grafo este hecho para que se pueda llegar a cada nodo desde cualquier nodo, se logra agregando el nodo actual a todos sus nodos adyacentes, esto es una actualización del código generador de grafos que se usó en la actividad 4.

Por ejemplo, si se le pide un grafo donde solo estén conectados a un nodo, ANTES se generaban grafos donde algunas rutas eran imposibles (en la imagen 1 no se puede ir del nodo 4 al nodo 1) AHORA todos los nodos están conectados de ida y vuelta (en la imagen 2 la ruta del nodo 4 al nodo 1 es posible)

Antes de la actualización

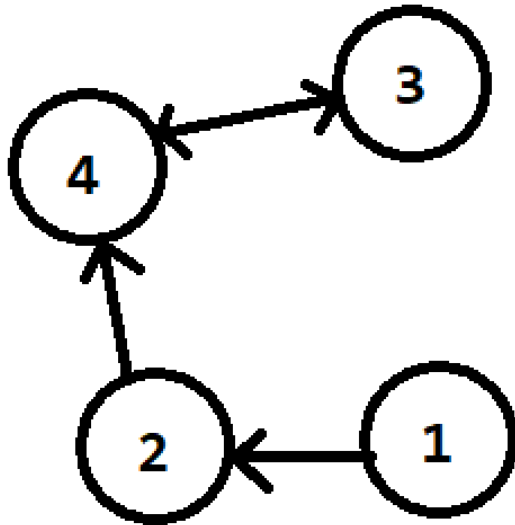


Imagen 5

Después de la actualización

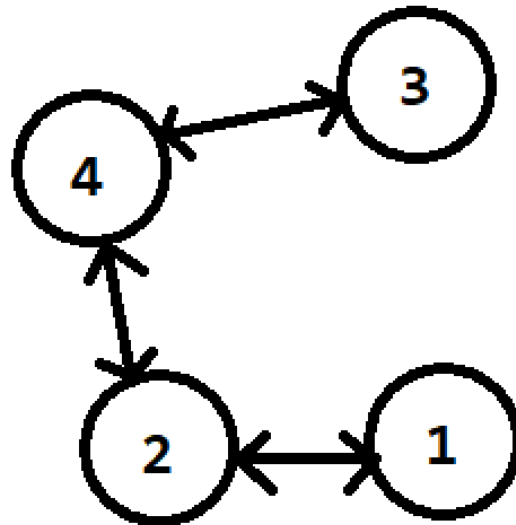


Imagen 6

Nota: en las imágenes de grafos siguientes las líneas que unen a los nodos no tendrán flecha, eso para indicar que cada arista es de ida y vuelta, eso es lo que hacer la actualización

FUNCIÓN DE COSTO

La función de costo esta dada por un rango aleatorio entre mínimo 2 y máximo 30.

APLICACIÓN EN LA VIDA REAL

Encontrar la ruta más corta en una ciudad es necesario para ahorrar tiempo y recursos como gasolina y electricidad, Esto se aplica en sistemas de GPS.

SECCION 2. DESCRIPCION DE METODOS DE SOLUCION

Para solucionar el problema usare 3 algoritmos:

1. Búsqueda por anchura (no informada)
2. Búsqueda voraz primero el mejor (informada)
3. Recocido simulado (meta-heurística)

BUSQUEDA POR ANCHURA (BPA)

La búsqueda primero en anchura es una estrategia sencilla en la que se expande primero el nodo raíz, a continuación, se expanden todos los sucesores del nodo raíz, después sus sucesores, etc. En general, se expanden todos los nodos a una profundidad en el árbol de búsqueda antes de expandir cualquier nodo del próximo nivel.

PSEUDOCODIGO

Mientras no encuentre al nodo final, visito los hijos del nodo inicial, al terminar visito los hijos del primer hijo del nodo inicial, al terminar visito los hijos del segundo hijo del nodo inicial, así sucesivamente hasta encontrar al nodo final.

BÚSQUEDA VORAZ PRIMERO EL MEJOR (BVPM)

Trata de expandir el nodo más cercano al objetivo, alegando que a lo mejor conduzca rápidamente a una solución. Así, evalúa los nodos utilizando solamente la función heurística: $f(n)=h(n)$.

PSEUDOCODIGO

Mientras no encuentre al nodo final, visito los hijos del nodo inicial, el nodo actual será el que tenga una menor distancia, ahora visito los hijos del nodo actual, así hasta que encuentre el nodo final .

RECOCIDO SIMULADO (RS)

Simulated annealing (SA), también llamado recocido simulado, cristalización simulada o enfriamiento simulado, es un 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. Dicho "óptimo global" corresponde a la solución del problema de interés para el que no existe un mejor valor. En el caso de que tal problema sea de minimización, el óptimo global será aquél para el cual la función objetivo tenga el más pequeño posible de todos los de su ([[espacio de de busqueda]]). Por el contrario, para un problema de maximización, el óptimo global es aquél con el valor más alto posible.

PSEUDOCODIGO

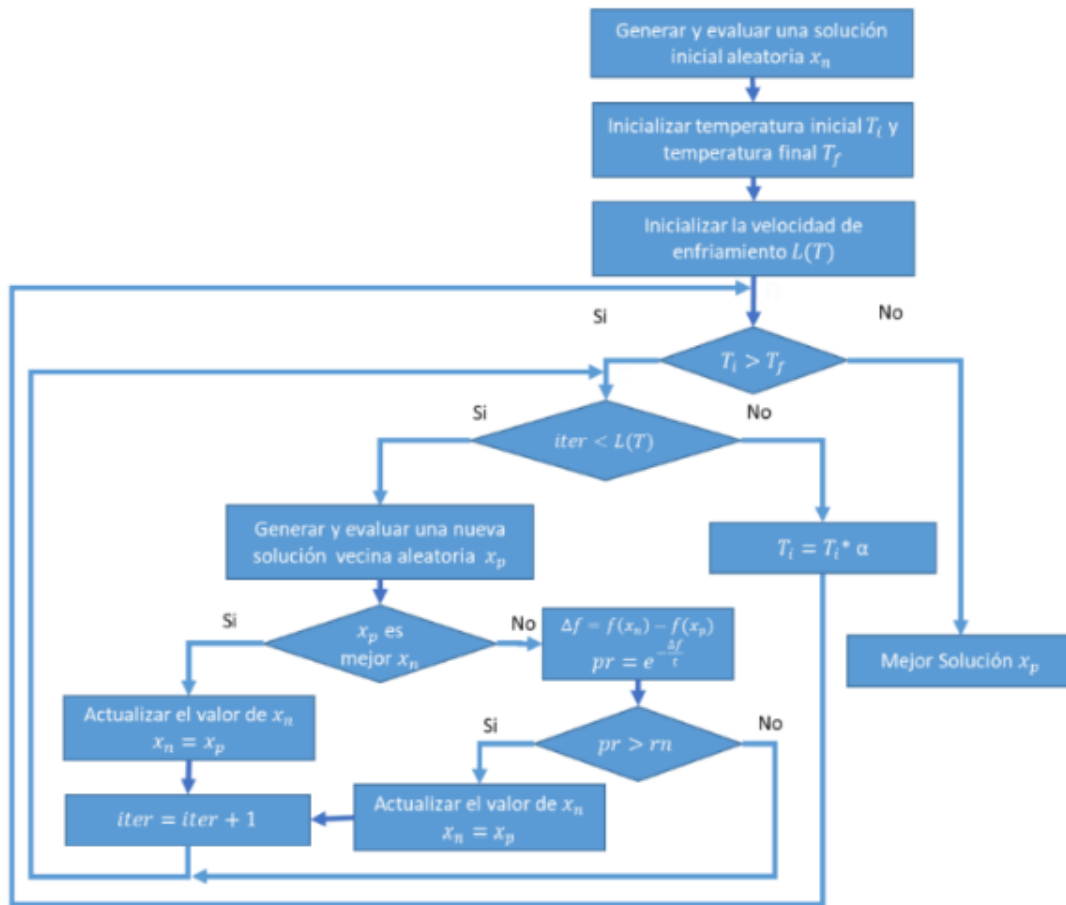


Imagen 7

SECCION 3. EXPERIMENTACION

DESCRIPCIÓN DE LAS INSTANCIAS

Habr  2 tipos de instancias diferentes, cada tipo tendr  10 instancias distintas generadas por el c digo creador de grafos lo cual hace que cada instancia (sin importar el tipo) sea  nica.

Las instancias de tipo 1 tendr n como caracter stica la variaci n de la cantidad de nodos adyacentes por nodo y la cantidad de nodos ser  fija para todas las 10 instancias generadas, la cantidad de nodos ser  de 500 (una estimaci n pensada debido a que en la actividad anterior con 100 nodos lo resolv a demasiado r pido).

Ejemplo de instancias de tipo 1

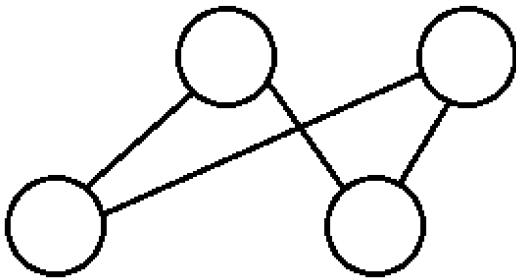


Imagen 8. Grafo con 2 nodos adyacentes por cada nodo

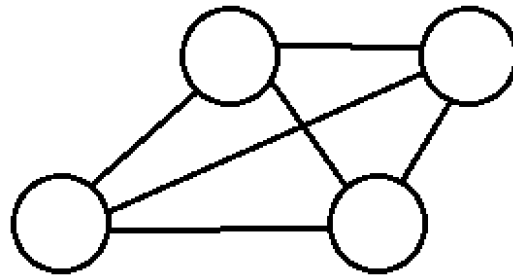


Imagen 9. Grafo con 3 nodos adyacentes por cada nodo

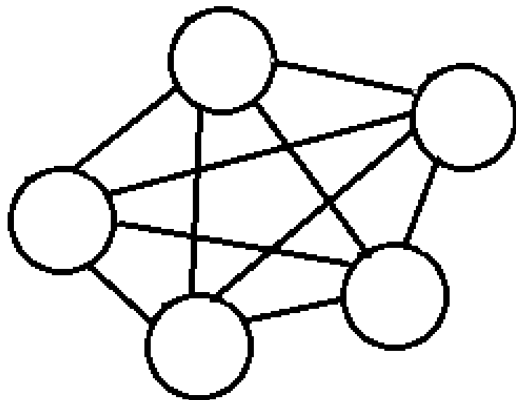


Imagen 10. Grafo con 4 nodos adyacentes por cada nodo

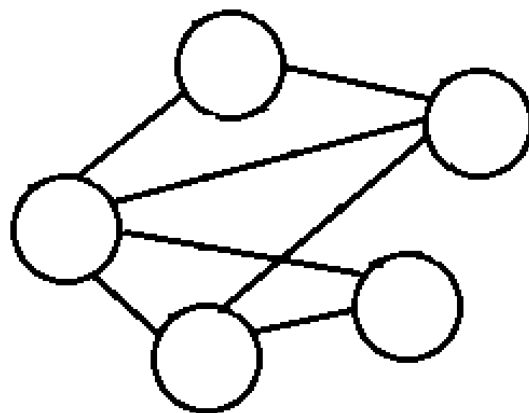


Imagen 11. Grafo con rango de nodos adyacentes para cada nodo m ximo 4 m nimo 2 (no se usaron este tipo de grafos)

NOTA: La cantidad de nodos que se muestra es para ejemplificar, Las instancias de prueba tendrán 500 nodos.

Las instancias de tipo 2 tendrán como característica la variación en la cantidad de nodos, este tipo se dividirá en 2 subgrupos, el primer subgrupo tendrá el 5 % de nodos adyacentes de la cantidad de nodos del grafo, por ejemplo en un grafo de 100 nodos, cada nodo de ese grafo tendrá 5 nodo adyacentes, en el caso del grafo con 1000 nodos cada nodo tendrá 50, el segundo subgrupo tendrá una cantidad fija de nodos adyacentes sin importar la cantidad de nodos del grafo, la cantidad fija será de 35, ejemplo un grafo de 100 nodos tendrá para cada nodo 35 nodos adyacentes, en un grafo de 1000 cada nodo tendrá 35 nodos adyacentes.

Ejemplos de instancias de tipo 2

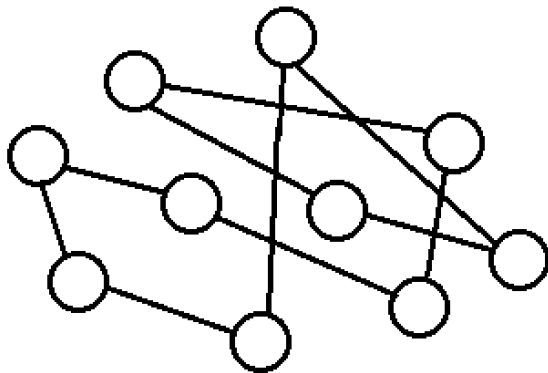


Imagen 12. Grafo con 10 nodos, cada nodo con 2 nodos adyacentes (20%), este grafo es del subtipo 1

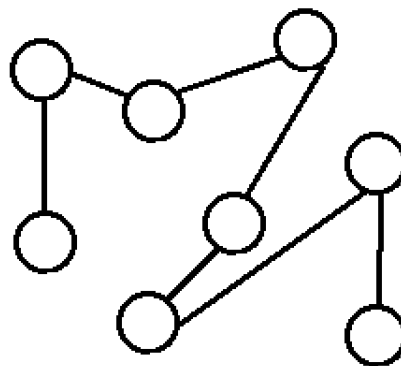


Imagen 13. Grafo con 8 nodos, cada nodo con 2 nodos adyacentes, si fuera un grafo de 12 nodos la cantidad de nodos adyacentes sería la misma, este grafo es del subtipo 2

NOTA: Los porcentajes y número de las imágenes 12 y 13 son con fines ilustrativos ya que los grafos con números y porcentajes reales serían muy difíciles de dibujar

COMO SE CONSTRUYE UNA INSTANCIA

A continuación se describe de manera breve como se construye un grafo con el código generador de grafos.

La función generadora necesita de 4 parámetros iniciales

```
def inst_ruta_mas_corta(cantidadNodos, conCosto, conDistancia, esConexo):
```

- 1) CantidadNodos: número de nodos que tendrá el grafo
- 2) conCosto: booleano, su valor habilita si el grafo tendrá costo entre la conexión de los nodos
- 3) conDistancia: booleano, su valor habilita si el grafo tendrá distancia entre la conexión de los nodos

- 4) esConexo: booleano, habilita si el grafo puede tener nodos sin salida o no

Ningún parámetro anterior excluye al otro, el único indispensable es cantidadNodos debe tener un valor mayor a 2.

Para este trabajo, todas las instancias tendrán:

- 1) CantidadNodos mayor a 2
- 2) conCosto igual a falso
- 3) ConDistancia igual a verdadero
- 4) esConexo igual a verdadero

Lo que hace el código es crear los nodos que pedimos, empezando por el nodo 1, apunta a X cantidad de nodos, al hacerlo genera el valor de la distancia, una vez todos los nodos apuntaron a algún nodo, se valida que todos los caminos tengan ida y vuelta, si un camino esta incompleto se modifica para que este correcto.

INSTANCIAS USADAS PARA ESTA EXPERIMENTACIÓN

Las instancias de tipo 1 tendrán 500 nodos

Ruta para buscar en las instancias de tipo 1 es:

- Ruta nodo 100 al nodo 400

Las instancias de tipo 2, subtipo 1 y 2 tendrán 200,400,600,800 y 1000 nodos

Rutas para buscar en las instancias de tipo 2 son:

- Instancias 200n10a (5%) y 200n20a (f) se buscará la ruta del nodo 50 a 150
- Instancias 400n20a (5%) y 400n20a (f) se buscará ruta del nodo 50 a 350
- Instancias 600n30a (5%) y 600n20a (f) se buscará la ruta del nodo 50 a 550
- Instancias 800n40a (5%) y 800n20a (f) se buscará la ruta del nodo 50 a 750
- Instancias 1000n50a (5%) y 1000n20a (f) se buscará la ruta del nodo 50 a 950

QUE SE ESPERA DEL EXPERIMENTO

Lo que se espera observar en la experimentación es si afecta el tiempo de cómputo los siguientes parámetros del grafo:

- El número de nodos que tiene el grafo
- El número de nodos adyacentes por nodo

HIPÓTESIS

Creo que el numero de nodos adyacentes por nodo si afectara al tiempo de cómputo elevándolo, el algoritmo que más sufrirá este efecto en primer lugar es el recocido y en segundo el de búsqueda por anchura.

La cantidad de nodos no considero que afecte mucho en el tiempo de cómputo a ningún algoritmo, lo que considero que variaría seria la respuesta mas optima dada por el algoritmo de recocido porque a mayor numero de nodos puede que los mínimos locales sean muchos y dependiendo de la temperatura no le daría el tiempo a salir de muchos de ellos.

SECCION 4. TABLAS Y GRAFICAS

Tablas probando las instancias de tipo 1

Numero de prueba	Algoritmo	Instancia	TIEMPO	SOLUCION	NUMERO DE NODOS
0	BVPM	500n20a	0.0	19	6
1	BVPM	500n20a	0.0	19	6
2	BVPM	500n20a	0.0	19	6
3	BVPM	500n20a	0.0	19	6
4	BVPM	500n20a	0.0	19	6
5	BPA	500n20a	0.022507190704345703	39	3
6	BPA	500n20a	0.01600170135498047	39	3
7	BPA	500n20a	0.017000675201416016	39	3
8	BPA	500n20a	0.016001224517822266	39	3
9	BPA	500n20a	0.018001079559326172	39	3
10	RS	500n20a	59.158467292785645	96	11
11	RS	500n20a	73.29352951049805	23	4
12	RS	500n20a	68.99218130111694	332	22
13	RS	500n20a	62.68930768966675	46	4
14	RS	500n20a	76.75280618667603	110	10
15	BVPM	500n40a	0.0	13	5
16	BVPM	500n40a	0.0	13	5
17	BVPM	500n40a	0.0	13	5
18	BVPM	500n40a	0.0	13	5
19	BVPM	500n40a	0.0	13	5
20	BPA	500n40a	0.011392831802368164	21	3
21	BPA	500n40a	0.004002809524536133	21	3
22	BPA	500n40a	0.0029964447021484375	21	3
23	BPA	500n40a	0.004000425338745117	21	3
24	BPA	500n40a	0.003000497817993164	21	3
25	RS	500n40a	140.9135308265686	175	12
26	RS	500n40a	121.96988320350647	21	3
27	RS	500n40a	130.30301785469055	88	6
28	RS	500n40a	112.97937798500061	64	5
29	RS	500n40a	107.71954941749573	126	10
30	BVPM	500n60a	0.0	19	2
31	BVPM	500n60a	0.0	19	2
32	BVPM	500n60a	0.0	19	2
33	BVPM	500n60a	0.0	19	2
34	BVPM	500n60a	0.00	19	2
35	BPA	500n60a	0.0019996166229248047	19	2
36	BPA	500n60a	0.0019991397857666016	19	2
37	BPA	500n60a	0.0010001659393310547	19	2
38	BPA	500n60a	0.0010006427764892578	19	2
39	BPA	500n60a	0.0009996891021728516	19	2
40	RS	500n60a	197.9529263973236	19	2
41	RS	500n60a	179.6658902168274	19	2

42	RS	500n60a	206.90477800369263	19	2
43	RS	500n60a	196.23641729354858	41	3
44 ¹	RS	500n60a	181.9148404598236	0	0
45	BVPM	500n80a	0.0	16	4
46	BVPM	500n80a	0.0	16	4
47	BVPM	500n80a	0.0	16	4
48	BVPM	500n80a	0.0	16	4
49	BVPM	500n80a	0.0	16	4
50	BPA	500n80a	0.6626439094543457	24	3
51	BPA	500n80a	0.6551616191864014	24	3
52	BPA	500n80a	0.6527771949768066	24	3
53	BPA	500n80a	0.6690618991851807	24	3
54	BPA	500n80a	0.5520691871643066	24	3
55	RS	500n80a	231.00981950759888	102	5
56	RS	500n80a	248.59121108055115	62	4
57	RS	500n80a	247.84243035316467	33	3
58	RS	500n80a	245.24169301986694	103	9
59	RS	500n80a	273.55981373786926	209	11
60	BVPM	500n100a	0.0	23	3
61	BVPM	500n100a	0.0	23	3
62	BVPM	500n100a	0.0	23	3
63	BVPM	500n100a	0.0	23	3
64	BVPM	500n100a	0.0	23	3
65	BPA	500n100a	0.028771162033081055	14	3
66	BPA	500n100a	0.025002241134643555	14	3
67	BPA	500n100a	0.027001380920410156	14	3
68	BPA	500n100a	0.026002168655395508	14	3
69	BPA	500n100a	0.026001691818237305	14	3
70	RS	500n100a	349.7299551963806	34	5
71	RS	500n100a	269.200475692749	80	5
72	RS	500n100a	320.06236839294434	42	5
73	RS	500n100a	332.8859086036682	103	11
74	RS	500n100a	299.9974319934845	54	5

Tabla 1. Resultados de ejecutar los algoritmos BVPM, BPA y RS cinco veces cada uno en grafos con distinta cantidad de nodos adyacentes.

[1] En la prueba 44 el algoritmo RS no pudo encontrar una solución antes de que se terminara la temperatura, cuando pasa eso devuelve una solución de 0.

Tablas probando las instancias de tipo 2 subtipo 1 (5%)

Numero de prueba	Algoritmo	Instancia	TIEMPO	SOLUCION	NUMERO DE NODOS
75	BVPM	200n10a (5%)	0.0	26	8
76	BVPM	200n10a (5%)	0.0	26	8
77	BVPM	200n10a (5%)	0.0	26	8
78	BVPM	200n10a (5%)	0.0	26	8

79	BVPM	200n10a (5%)	0.0	26	8
80	BPA	200n10a (5%)	0.0	38	3
81	BPA	200n10a (5%)	0.0	38	3
82	BPA	200n10a (5%)	0.0	38	3
83	BPA	200n10a (5%)	0.0	38	3
84	BPA	200n10a (5%)	0.0	38	3
85	RS	200n10a (5%)	5.619655132293701	38	3
86	RS	200n10a (5%)	5.797254800796509	107	10
87	RS	200n10a (5%)	6.9693522453308105	93	6
88	RS	200n10a (5%)	5.141760349273682	38	3
89	RS	200n10a (5%)	6.886631965637207	103	7
90	BVPM	400n20a (5%)	0.0	35	12
91	BVPM	400n20a (5%)	0.00100302696228027	35	12
92	BVPM	400n20a (5%)	0.0	35	12
93	BVPM	400n20a (5%)	0.00099730491638184	35	12
94	BVPM	400n20a (5%)	0.0	35	12
95	BPA	400n20a (5%)	0.01574230194091797	34	3
96	BPA	400n20a (5%)	0.0130023956298828	34	3
97	BPA	400n20a (5%)	0.01299929618835449	34	3
98	BPA	400n20a (5%)	0.01300215721130371	34	3
99	BPA	400n20a (5%)	0.01399993896484375	34	3
100	RS	400n20a (5%)	45.95292782783508	79	6
101	RS	400n20a (5%)	50.573007583618164	80	6
102	RS	400n20a (5%)	46.96527981758118	15	3
103	RS	400n20a (5%)	43.59386372566223	126	12
104	RS	400n20a (5%)	47.50337266921997	115	8
105	BVPM	600n30a (5%)	0.0	26	9
106	BVPM	600n30a (5%)	0.0	26	9
107	BVPM	600n30a (5%)	0.0	26	9
108	BVPM	600n30a (5%)	0.0	26	9
109	BVPM	600n30a (5%)	0.0	26	9
110	BPA	600n30a (5%)	0.03400230407714844	40	3
111	BPA	600n30a (5%)	0.02800321578979492	40	3
112	BPA	600n30a (5%)	0.0280010700225831	40	3
113	BPA	600n30a (5%)	0.0280034542083740	40	3
114	BPA	600n30a (5%)	0.02900218963623047	40	3
115	RS	600n30a (5%)	125.29951643943787	129	11
116	RS	600n30a (5%)	139.6850323677063	144	9
117	RS	600n30a (5%)	135.92392706871033	46	4
118	RS	600n30a (5%)	122.81941485404968	96	6
119	RS	600n30a (5%)	139.80853271484375	95	8
120	BVPM	800n40a (5%)	0.00503563880920410	82	29
121	BVPM	800n40a (5%)	0.00339365005493164	82	29
122	BVPM	800n40a (5%)	0.00200080871582031	82	29
123	BVPM	800n40a (5%)	0.00200176239013671	82	29
124	BVPM	800n40a (5%)	0.0019993782043457	82	29
125	BPA	800n40a (5%)	0.55419087409997314	17	3
126	BPA	800n40a (5%)	0.5065493583679199	17	3

127	BPA	800n40a (5%)	0.5075399875640869	17	3
128	BPA	800n40a (5%)	0.507124662399292	17	3
129	BPA	800n40a (5%)	0.5041036605834961	17	3
130	RS	800n40a (5%)	350.0950026512146	74	6
131	RS	800n40a (5%)	326.58192205429077	173	14
132	RS	800n40a (5%)	370.7462887763977	270	17
133	RS	800n40a (5%)	335.25698590278625	79	7
134	RS	800n40a (5%)	343.7224371433258	176	12
135	BVPM	1000n50a (5%)	0.00201964378356933	47	16
136	BVPM	1000n50a (5%)	0.00120401382446289	47	16
137	BVPM	1000n50a (5%)	0.00099873542785644	47	16
138	BVPM	1000n50a (5%)	0.00099563598632812	47	16
139	BVPM	1000n50a (5%)	0.00200128555297851	47	16
140	BPA	1000n50a (5%)	0.03645658493041992	40	3
141	BPA	1000n50a (5%)	0.03300285339355469	40	3
142	BPA	1000n50a (5%)	0.03400206565856933	40	3
143	BPA	1000n50a (5%)	0.03600668907165527	40	3
144	BPA	1000n50a (5%)	0.03299927711486816	40	3
145	RS	1000n50a (5%)	714.3437449932098	184	13
146	RS	1000n50a (5%)	718.2001826763153	63	5
147	RS	1000n50a (5%)	672.9857180118561	634	37
148	RS	1000n50a (5%)	720.3598303794861	236	15
149	RS	1000n50a (5%)	627.837904214859	79	7

Tabla 2. Resultados de ejecutar Los algoritmos BVPM, BPA y RS cinco veces cada uno en grafos con el mismo porcentaje de nodos adyacentes pero la cantidad de nodos del grafo variando.

Tablas probando las instancias de tipo 2 subtipo 2 (20 adyacentes fijo)

Numero de prueba	Algoritmo	Instancia	TIEMPO	SOLUCION	NUMERO DE NODOS
150	BVPM	200n20a (f)	0.0	27	2
151	BVPM	200n20a (f)	0.0	27	2
152	BVPM	200n20a (f)	0.0	27	2
153	BVPM	200n20a (f)	0.0	27	2
154	BVPM	200n20a (f)	0.0	27	2
155	BPA	200n20a (f)	0.0	27	2
156	BPA	200n20a (f)	0.0	27	2
157	BPA	200n20a (f)	0.0	27	2
158	BPA	200n20a (f)	0.0	27	2
159	BPA	200n20a (f)	0.0	27	2
160	RS	200n20a (f)	8.786673545837402	27	2
161	RS	200n20a (f)	8.71998643875122	27	3
162	RS	200n20a (f)	9.353179216384888	13	4
163	RS	200n20a (f)	8.884111404418945	27	2
164	RS	200n20a (f)	8.6339111328125	19	3

165	BVPM	400n20a (f)	0.0	16	6
166	BVPM	400n20a (f)	0.0	16	6
167	BVPM	400n20a (f)	0.0	16	6
168	BVPM	400n20a (f)	0.0	16	6
169	BVPM	400n20a (f)	0.0	16	6
170	BPA	400n20a (f)	0.1945490837097168	49	3
171	BPA	400n20a (f)	0.1870133876800537	49	3
172	BPA	400n20a (f)	0.18691778182983398	49	3
173	BPA	400n20a (f)	0.22801685333251953	49	3
174	BPA	400n20a (f)	0.18601369857788086	49	3
175	RS	400n20a (f)	44.418500661849976	174	11
176	RS	400n20a (f)	45.25475192070007	48	6
177	RS	400n20a (f)	43.74787783622742	103	8
178	RS	400n20a (f)	37.200814723968506	60	5
179	RS	400n20a (f)	42.528815269470215	14	3
180	BVPM	600n20a (f)	0.0	36	13
181	BVPM	600n20a (f)	0.0	36	13
182	BVPM	600n20a (f)	0.0	36	13
183	BVPM	600n20a (f)	0.0	36	13
184	BVPM	600n20a (f)	0.0	36	13
185 ²	BPA	600n20a (f)	0.048596858978271484	46	3
186 ²	BPA	600n20a (f)	0.04100298881530762	46	3
187 ²	BPA	600n20a (f)	0.04200267791748047	46	3
188 ²	BPA	600n20a (f)	0.04100370407104492	46	3
189 ²	BPA	600n20a (f)	0.042003631591796875	46	3
190	RS	600n20a (f)	84.53110122680664	76	4
191	RS	600n20a (f)	82.94886755943298	101	7
192	RS	600n20a (f)	93.5615382194519	300	16
193	RS	600n20a (f)	86.88771653175354	88	6
194	RS	600n20a (f)	95.95267939567566	127	10
195	BVPM	800n20a (f)	0.0030014514923095703	149	62
196	BVPM	800n20a (f)	0.004001140594482422	149	62
197	BVPM	800n20a (f)	0.0019981861114501953	149	62
198	BVPM	800n20a (f)	0.003000497817993164	149	62
199	BVPM	800n20a (f)	0.002000570297241211	149	62
200	BPA	800n20a (f)	0.04700207710266113	36	3
201	BPA	800n20a (f)	0.04800581932067871	36	3
202	BPA	800n20a (f)	0.04500079154968262	36	3
203	BPA	800n20a (f)	0.04500412940979004	36	3
204	BPA	800n20a (f)	0.0440065860748291	36	3
205	RS	800n20a (f)	145.4641797542572	140	8
206	RS	800n20a (f)	137.2003619670868	115	8
207	RS	800n20a (f)	148.6063141822815	205	13
208	RS	800n20a (f)	149.41038918495178	36	3
209	RS	800n20a (f)	151.73344087600708	60	4
210	BVPM	1000n20a (f)	0.002983570098876953	71	25
211	BVPM	1000n20a (f)	0.0009946823120117188	71	25
212	BVPM	1000n20a (f)	0.0013706684112548828	71	25

213	BVPM	1000n20a (f)	0.001001119613647461	71	25
214	BVPM	1000n20a (f)	0.0019996166229248047	71	25
215	BPA	1000n20a (f)	0.01299905776977539	58	3
216	BPA	1000n20a (f)	0.01200103759765625	58	3
217	BPA	1000n20a (f)	0.012003660202026367	58	3
218	BPA	1000n20a (f)	0.011998176574707031	58	3
219	BPA	1000n20a (f)	0.012001276016235352	58	3
220	RS	1000n20a (f)	322.8989541530609	665	40
221	RS	1000n20a (f)	303.0197117328644	112	8
222	RS	1000n20a (f)	276.03368973731995	67	4
223	RS	1000n20a (f)	286.21514201164246	398	27
224	RS	1000n20a (f)	285.97946071624756	213	13

Tabla 3. Resultados de ejecutar los algoritmos BVPM, BPA y RS cinco veces cada uno en grafos con la misma cantidad de nodos, pero con la cantidad de nodos del grafo variando.

[2] En las pruebas del número 185 al 189 la ruta a buscar cambio por “nodo inicial 50 al nodo final 151”, al probarlo con la ruta anterior no dio una solución en un tiempo razonable (más de 15 minutos en ejecución sin resultado), decidí cambiar la ruta ya que en pruebas anteriores el tiempo de ejecución no pasaba de 1 segundo

PROMEDIOS DEL ALGORITMO BÚSQUEDA VORAZ PRIMERO EL MEJOR (BVPM)

INSTANCIA	TIEMPO	SOLUCION	NODOS
500N20A	0.0	19	6
500N40A	0.0	13	5
500N60A	0.0	19	2
500N80A	0.0	16	4
500N100A	0.0	23	3

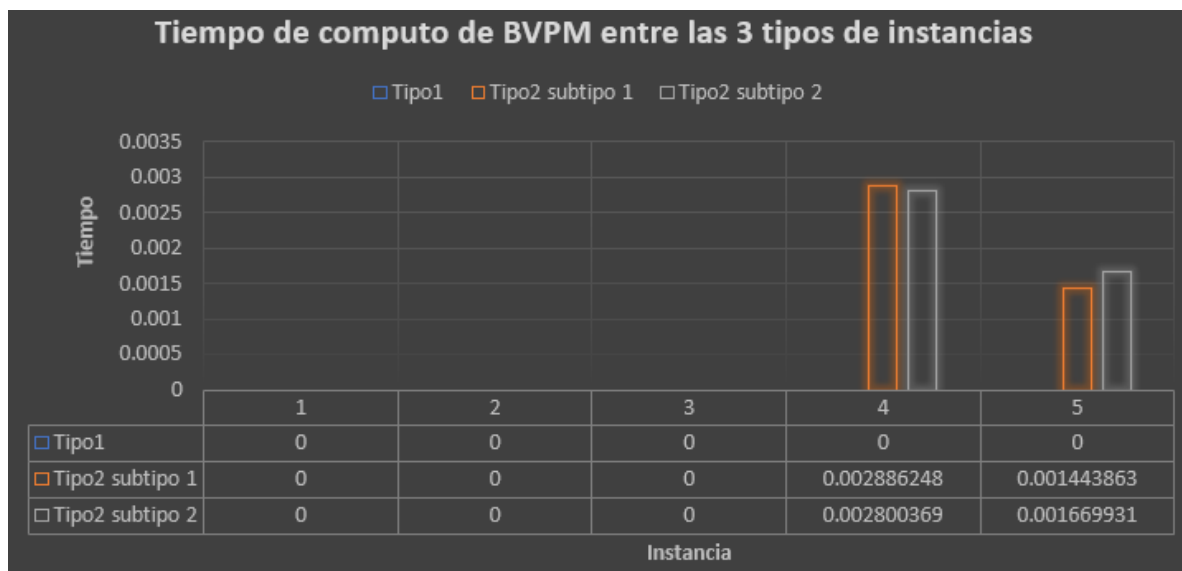
Tabla 4. Resultados promedios del algoritmo BVPM con las instancias de tipo 1 (instancias con cantidad de nodos fijo “500” y cantidad de nodos adyacentes variable “20,40,60,80 y 100 nodos”)

INSTANCIA	TIEMPO	SOLUCION	NODOS
200N10A (5%)	0.0	26	8
400N20A (5%)	0.0	35	12
600N30A (5%)	0.0	26	9
800N40A (5%)	0.002886248	82	29
1000N50A (5%)	0.001443863	47	16

Tabla 5. Resultados promedios del algoritmo BVPM con las instancias de tipo 2, subtipo 1 (instancias con cantidad de nodos variable “200,400,600,800 y 1000” con “10,20,30,40,50” de nodos adyacentes respectivamente).

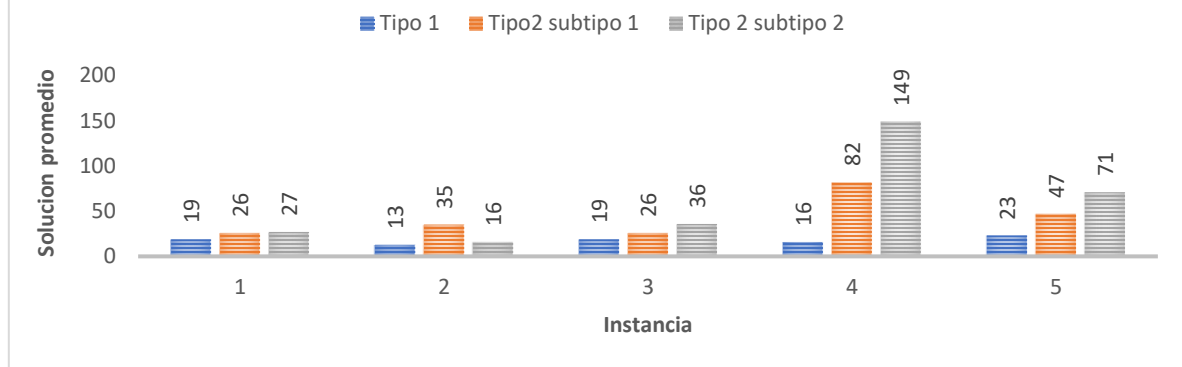
INSTANCIA	TIEMPO	SOLUCION	NODOS
200N20A (F)	0.0	27	2
400N20A (F)	0.0	16	6
600N20A (F)	0.0	36	13
800N20A (F)	0.002800369	149	62
1000N20A (F)	0.001669931	71	25

Tabla 6. Resultados promedios del algoritmo BVPM con las instancias de tipo 2, subtipo 2 (instancias con “200,400,600,800 y 1000” nodos todos con 20 nodos adyacentes).



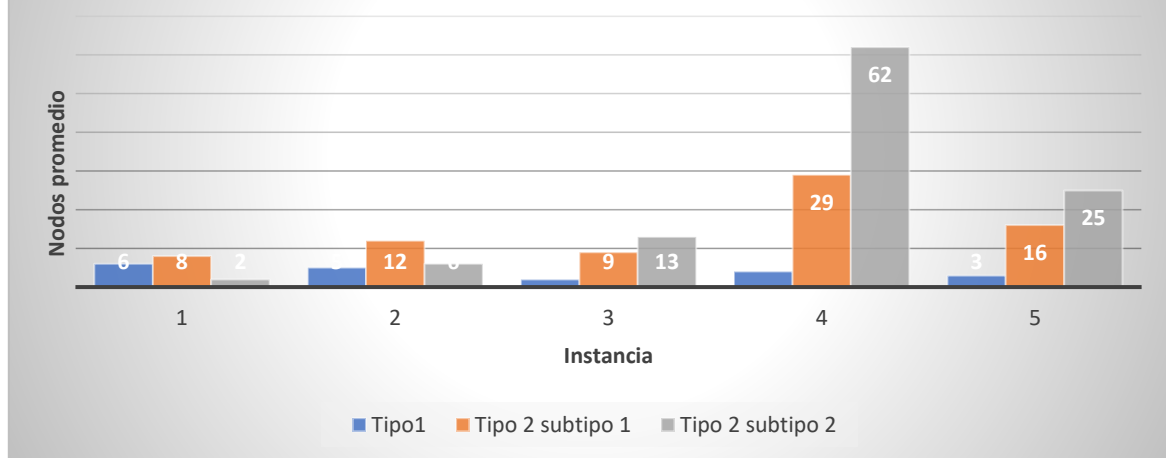
Grafica1. Tiempo promedio de cómputo BVPM al probar las instancias de tipo 1 y 2 de ambos subtipos.

SOLUCIONES DE BVPM ENTRE LAS 3 TIPOS DE INSTANCIAS



Grafica2. Soluciones promedio BVPM al probar las instancias de tipo 1 y 2 de ambos subtipos.

Cantidad de nodos de BVPM entre las 3 tipos de instancias



Grafica3. Nodos promedio BVPM al probar las instancias de tipo 1 y 2 de ambos subtipos.

PROMEDIOS DEL ALGORITMO BÚSQUEDA PRIMERO EN AMPLITUD (BPA)

INSTANCIA	TIEMPO	SOLUCION	NODOS
500N20A	0.017902374	39	3
500N40A	0.005078602	21	3
500N60A	0.001399851	19	2
500N80A	0.638342762	24	3
500N100A	0.026555729	14	3

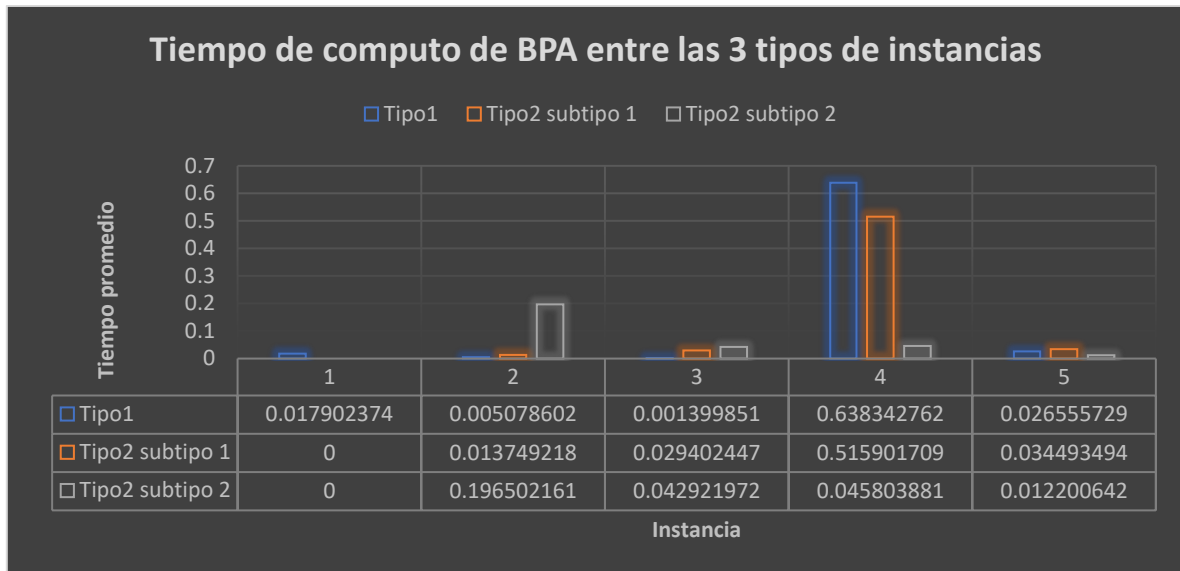
Tabla 7. Resultados promedios del algoritmo BPA con las instancias de tipo 1 (instancias con cantidad de nodos fijo "500" y cantidad de nodos adyacentes variable "20,40,60,80 y 100 nodos").

INSTANCIA	TIEMPO	SOLUCION	NODOS
200N10A (5%)	0.0	38	3
400N20A (5%)	0.013749218	34	3
600N30A (5%)	0.029402447	40	3
800N40A (5%)	0.515901709	17	3
1000N50A (5%)	0.034493494	40	3

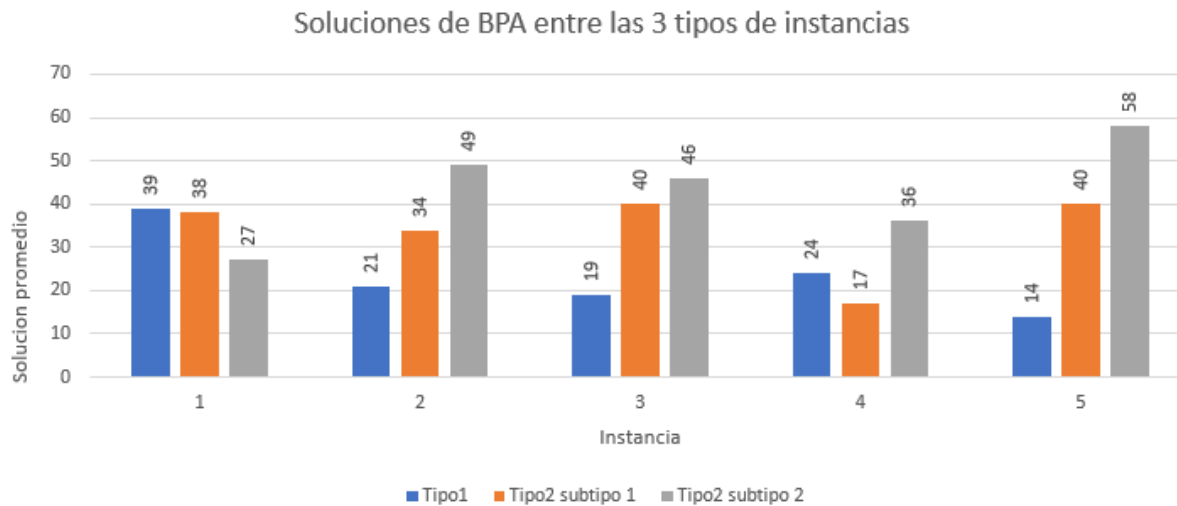
Tabla 8. Resultados promedios del algoritmo BPA con las instancias de tipo 2, subtipo 1 (instancias con cantidad de nodos variable "200,400,600,800 y 1000" con "10,20,30,40,50" de nodos adyacentes respectivamente).

INSTANCIA	TIEMPO	SOLUCION	NODOS
200N20A (F)	0.0	27	2
400N20A (F)	0.196502161	49	3
600N20A (F)	0.042921972	46	3
800N20A (F)	0.045803881	36	3
1000N20A (F)	0.012200642	58	3

Tabla 9. Resultados promedios del algoritmo BPA con las instancias de tipo 2, subtipo 2 (instancias con "200,400,600,800 y 1000" nodos todos con 20 nodos adyacentes).

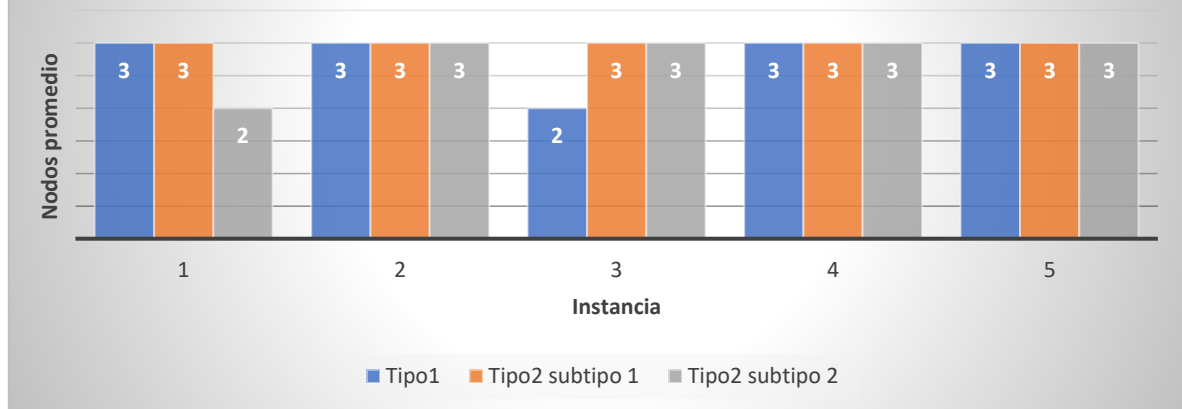


Grafica4. Tiempo promedio de cómputo BPA al probar las instancias de tipo 1 y 2 de ambos subtipos.



Grafica5. Soluciones promedio BPA al probar las instancias de tipo 1 y 2 de ambos subtipos.

Cantidad de nodos de BPA entre las 3 tipos de instancias



Grafica6. Nodos promedio BPA al probar las instancias de tipo 1 y 2 de ambos subtipos.

PROMEDIOS DEL ALGORITMO RECOCIDO SIMULADO (RS)

INSTANCIA	TIEMPO	SOLUCION	NODOS
500N20A	68.1772584	121.4	10.2
500N40A	122.7770719	94.8	7.2
500N60A	195.190003	24.5	2.25
500N80A	249.2489935	101.8	6.4
500N100A	314.375228	62.6	6.2

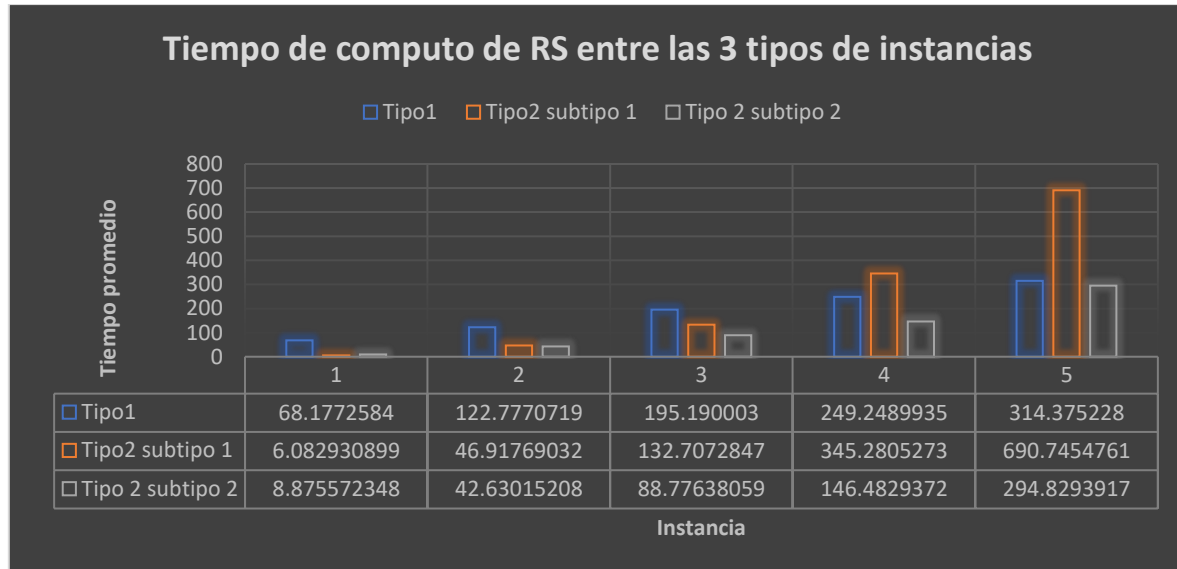
Tabla 10. Resultados promedios del algoritmo RS con las instancias de tipo 1 (instancias con cantidad de nodos fijo "500" y cantidad de nodos adyacentes variable "20,40,60,80 y 100 nodos").

INSTANCIA	TIEMPO	SOLUCION	NODOS
200N10A (5%)	6.082930899	75.8	5.8
400N20A (5%)	46.91769032	83	7
600N30A (5%)	132.7072847	102	7.6
800N40A (5%)	345.2805273	154.4	11.2
1000N50A (5%)	690.7454761	239.2	15.4

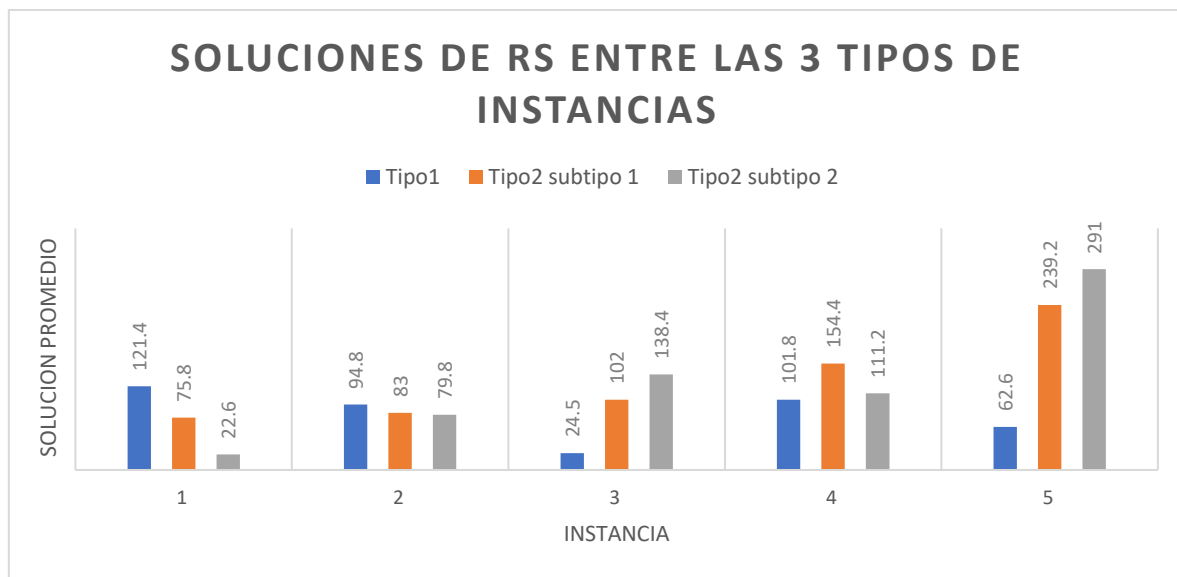
Tabla 11. Resultados promedios del algoritmo RS con las instancias de tipo 2, subtipo 1 (instancias con cantidad de nodos variable "200,400,600,800 y 1000" con "10,20,30,40,50" de nodos adyacentes respectivamente).

INSTANCIA	TIEMPO	SOLUCION	NODOS
200N20A (F)	8.875572348	22.6	2.8
400N20A (F)	42.63015208	79.8	6.6
600N20A (F)	88.77638059	138.4	8.6
800N20A (F)	146.4829372	111.2	7.2
1000N20A (F)	294.8293917	291	18.4

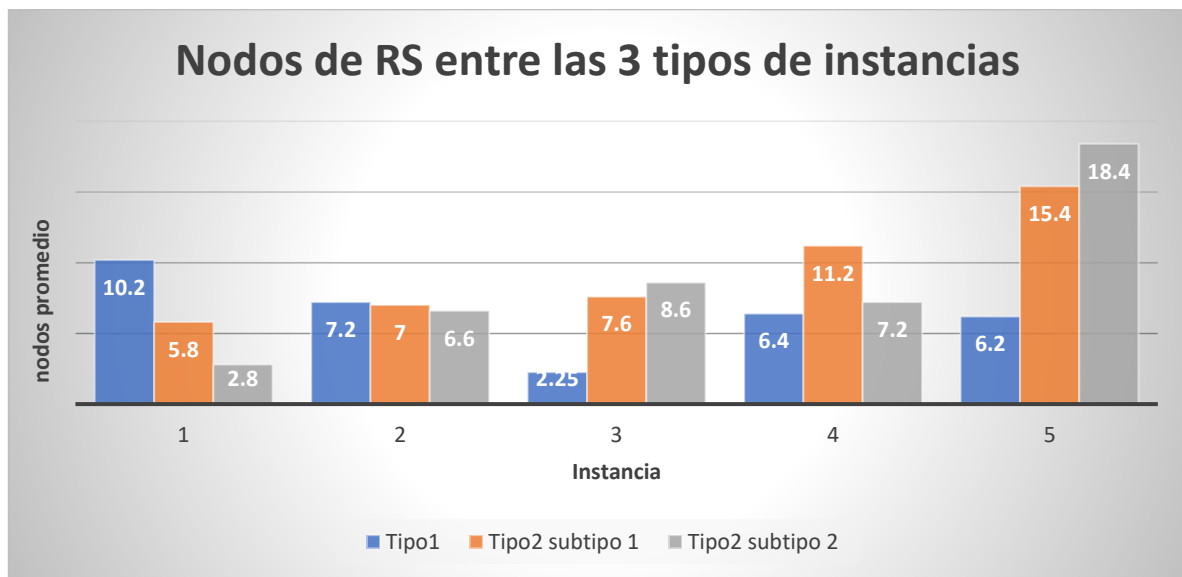
Tabla 12. Resultados promedios del algoritmo RS con las instancias de tipo 2, subtipo 2 (instancias con “200,400,600,800 y 1000” nodos todos con 20 nodos adyacentes).



Grafica7. Tiempo promedio de cómputo RS al probar las instancias de tipo 1 y 2 de ambos subtipos.



Grafica8. Soluciones promedio RS al probar las instancias de tipo 1 y 2 de ambos subtipos.



Grafica9. Nodos promedio RS al probar las instancias de tipo 1 y 2 de ambos subtipos.

INSTANCIAS

Tablas y graficas del generador de instancias

Numero de nodos	Cantidad de adyacentes	Tiempo de cómputo	Peso del archivo de texto generado
200	20	0.540513277053833	77 KB
400	20	2.133101463317871	161 KB
600	20	4.887146949768066	246 KB
800	20	9.067593812942505	331 KB
1000	20	14.64565134048462	415 KB

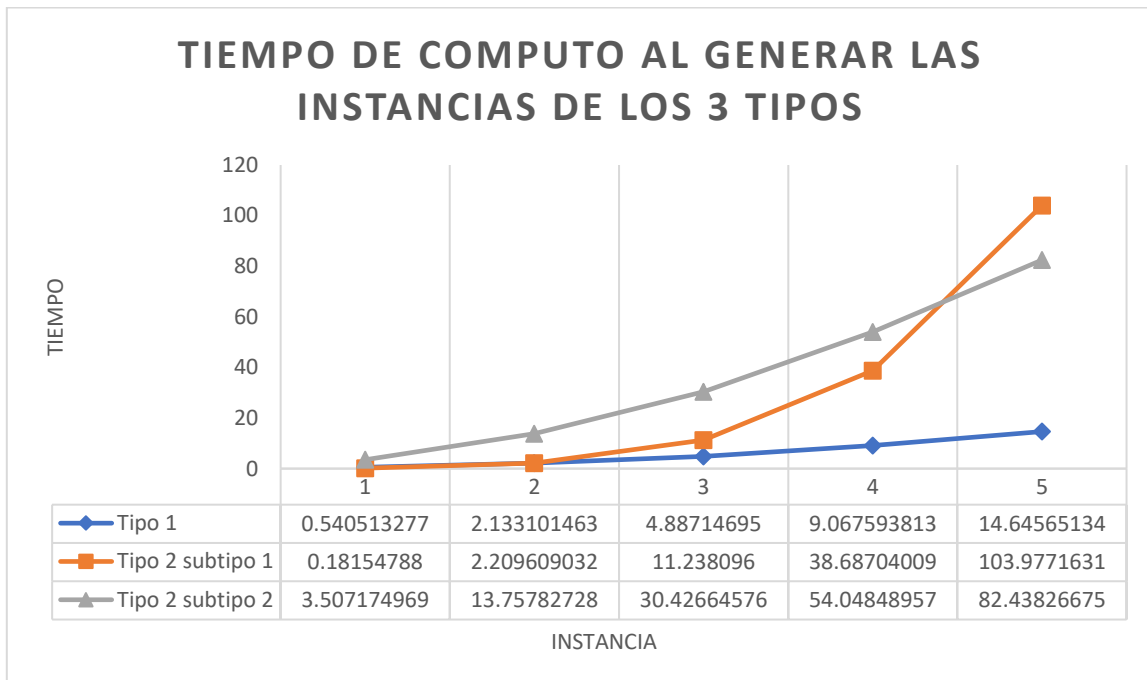
Grafica 13. Tiempo (en segundos) que tarda en generar las instancias con 20 nodos adyacentes (tipo 1).

Numero de nodos	Cantidad de adyacentes (5%)	Tiempo de cómputo	Peso del archivo de texto generado
200	10	0.1815478801727295	40 KB
400	20	2.209609031677246	161 KB
600	30	11.238095998764038	365 KB
800	40	38.68704009056091	650 KB
1000	50	103.97716307640076	1016 KB

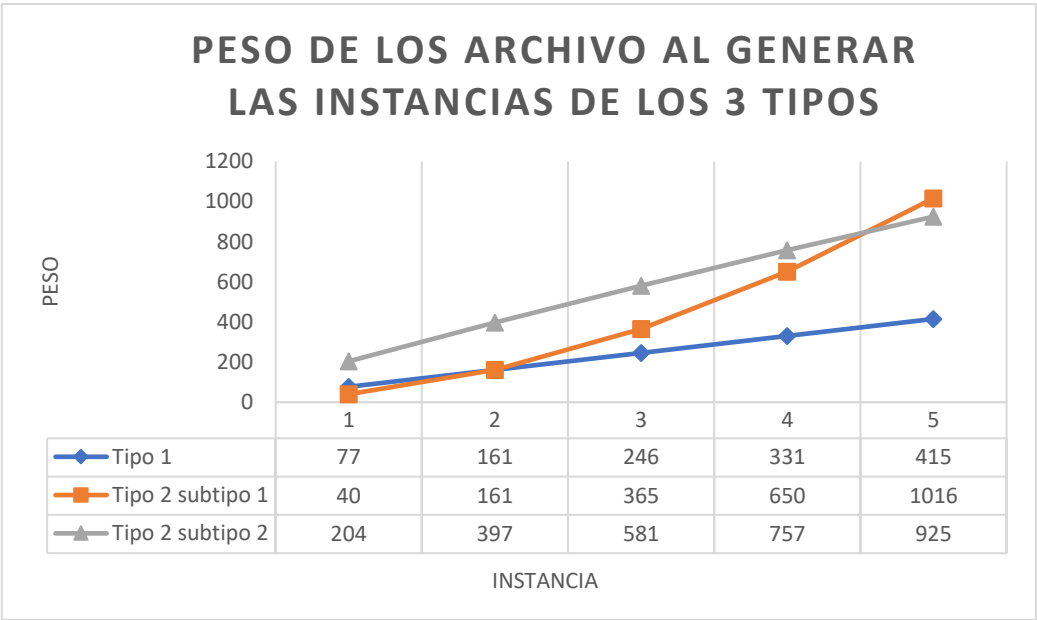
Grafica 14. Tiempo (en segundos) que tarda en generar las instancias con el 5% nodos adyacentes (tipo 2 subtipo 1).

Numero de nodos	Cantidad de adyacentes	Tiempo de cómputo	Peso del archivo de texto generado
500	20	3.5071749687194824	204 KB
500	40	13.757827281951904	397 KB
500	60	30.426645755767822	581 KB
500	80	54.048489570617676	757 KB
500	100	82.43826675415039	925 KB

Grafica 15. Tiempo (en segundos) que tarda en generar las instancias cada vez con más nodos adyacentes, pero con la misma cantidad de nodos (tipo 2 subtipo 2).



Grafica10. Tiempo (en segundos) de cómputo al generar cada instancia de todos los tipos



Grafica11. Peso (en KB) de Los archivos de cada instancia de todos Los tipos

SECCION 5. RESULTADOS Y CONCLUSIONES

Color	Significado
VERDE	Primer lugar, es el mejor parámetro o valor
NARANJA	Segundo lugar, no tal mal los resultados
ROJO	Tercer lugar, es el peor parámetro o valor

Tabla 16. Simbología

RESULTADOS ALGORITMO BVPM

	TIEMPO	SOLUCION	NODOS
TIPO 1	0.0	90	20
TIPO 2 SUBTIPO 1	0.004330111	216	74
TIPO 2 SUBTIPO 2	0.0044703	299	108

Tabla 17. suma de promedios de cada parámetro en cada tipo de instancia para el algoritmo BVPM

Conclusiones BVPM

1. El tiempo es el parámetro menos afectado por las instancias
2. Las instancias de tipo 1 es la que mejor pudo resolver y la peores fueron las instancias de tipo 2 subtipo 2

RESULTADOS ALGORITMO BPA

	TIEMPO	SOLUCION	NODOS
TIPO 1	0.689279318	117	14
TIPO 2 SUBTIPO 1	0.593546868	169	15
TIPO 2 SUBTIPO 2	0.29742866	216	14

Tabla 18. suma de promedios de cada parámetro en cada tipo de instancia para el algoritmo BPA

Conclusiones BPA

1. La cantidad de nodos fue el parámetro menos afectado por las instancias
2. Las instancias de tipo fue las mejores que pudo resolver y las peores las instancias de tipo 2 subtipo 2

RESULTADOS ALGORITMO RS

	TIEMPO	SOLUCION	NODOS
TIPO 1	949.7685548	405.1	32.25
TIPO 2 SUBTIPO 1	1221.733909	654.4	47
TIPO 2 SUBTIPO 2	581.594434	643	43.6

Tabla 19. suma de promedios de cada parámetro en cada tipo de instancia para el algoritmo RS

Conclusiones RS

1. Todos los parámetros fueron afectados por algún tipo de instancia
2. Las instancias de tipo 1 fueron las que mejor resolvió y las peores fueron las de tipo 2 subtipo 1

CONCLUSIONES SOBRE LAS INSTANCIAS

La instancia de tipo 1 fueron las que mejor pudieron resolver los 3 algoritmos, por lo que variar el numero de nodos adyacentes y mantener constante la cantidad de nodos del grafo es lo que menos afecto a los algoritmos.

Las instancias de tipo 2 fueron las que más dificultad presentaron a los 3 algoritmos, específicamente las de subtipo 2, por lo que aumentar la cantidad de nodos del grafo afecta a los algoritmos y mas aun si la cantidad de nodos adyacentes a un número fijo permanece constante.

RESULTADOS ENTRE ALGORITMOS

	TIEMPO	SOLUCION	NODOS
BVPM	0.0001013	40	13.46
BPA	0.105350	33.4666	2.86
RS	183.3627	113.17	8.16

Tabla 20. Tiempo, solución y nodos promedio de Los algoritmos BVPM (búsqueda voraz primero el mejor), BPA (búsqueda por amplitud) y RS (recocido simulado)

RESULTADOS DE LOS TIPOS DE INSTANCIA DE SOLUCION

	DE 0 A 25	DE 26 A 50	DE 50 A 500
BVPM	25	35	15
BPA	25	45	5

Tabla 21. Cantidad de soluciones entre esos rangos por Los algoritmos BVPM y BPA

CONCLUSIONES SOBRE LOS ALGORITMOS

Después de analizar esta tabla 20 y las gráficas 1 hasta la gráfica 9 se llegaron a las siguientes conclusiones:

- En primer lugar, quedo el algoritmo BPA, dando un tiempo bueno, con soluciones bajas y con muy pocos nodos.
- En segundo lugar, el algoritmo BVPM, dando el menor tiempo de cómputo, soluciones buenas, pero con una cantidad de nodos elevada.
- En tercer lugar, el algoritmo RS, fue el peor con un tiempo, solución y nodos muy elevados.

JUSTIFICACIÓN SOBRE LOS RESULTADOS DEL ALGORITMO RS

Considero que las soluciones malas que genero son porque los parámetros del algoritmo no le dieron el tiempo suficiente para llegar a la respuesta optima, no le di mejores parámetros porque el tiempo de cómputo se elevara demasiado incluso se tardaba mas de 30 minutos con la instancia más sencilla vista en este trabajo y para la cantidad de veces que tenía planeado probarlo se me haría imposible.

El motivo de los tiempos de cómputo elevados considero que son por una ineficiente implementación del algoritmo dado que el algoritmo RS necesita generar una solución vecina en mi caso yo genero una nueva solución por completo lo cual pienso que es lo que mas tiempo le cuesta, con mas tiempo para pensar se podría encontrar una mejor manera de generar vecino para este algoritmo.

Para este trabajo los parámetros del algoritmo RS son los siguientes:

parámetro	Valor
<i>Rn</i>	<i>0.1</i>
<i>Temperatura inicial</i>	<i>50</i>
<i>Temperatura final</i>	<i>30</i>
<i>Iteraciones máximas</i>	<i>100</i>
<i>Velocidad de enfriamiento</i>	<i>1</i>
<i>Diferencia de temperatura</i>	<i>20</i>

CODIGO

El código y los archivos de las instancias utilizadas para este proyecto se encuentra en este repositorio.

https://github.com/LUJAAA/PIA_IA.git

En los archivos del repositorio viene unas instrucciones para ejecutar los códigos y generar las instancias.