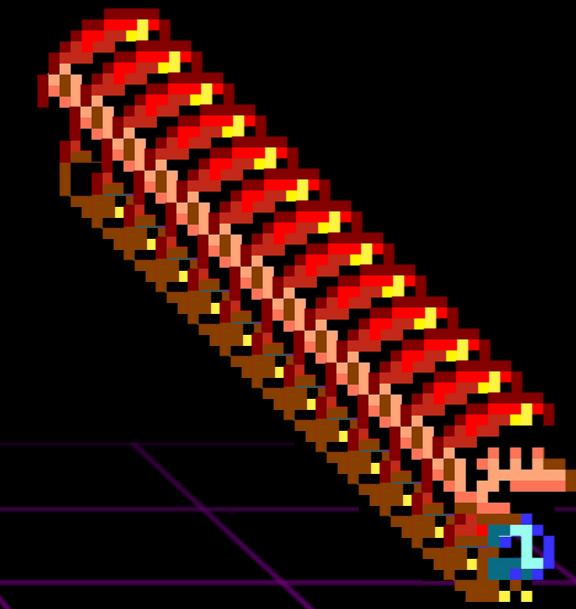


MÉTODOS CONSTRUCTIVOS Y ALEATORIZADOS

CAMILO
BERMÚDEZ

ALGORITMO CONSTRUCTIVO



CONSTRUYENDO EL GRAFO

Después de leer la entrada y calcular la distancia entre cada par de nodos, se arma un grafo en el cual cada nodo está conectado con los nodos a los cuales es factible que vaya (a estos se les llama vecinos).

CONSTRUYENDO EL GRAFO

¿Cómo sabemos que es factible que se vaya del nodo i al nodo j? Para esto, debe cumplirse:

- $\max(d_{0i}, e_i) + s_i + d_{ij} \leq l_j$
- $\max(d_{0i}, e_i) + s_i + d_{ij} + s_j + d_{j0} \leq l_0$
- $q_i + q_j \leq Q$

*la distancia euc. entre i y j es d_{ij}

CONSTRUYENDO EL GRAFO

A la arista entre el nodo i y el nodo j se le asigna un peso de acuerdo al criterio:

$$\alpha \cdot d_{ij} + (1 - \alpha) \cdot e_j + \mu \quad \forall i \neq 0$$

$$\beta \cdot d_{ij} - (1 - \beta) \cdot l_j \quad i = 0$$

CONSTRUYENDO EL GRAFO

$$\mu = \frac{d_{0i}}{\max(d_{0k})} \cdot \frac{d_{0j}}{\max(d_{0k})} \cdot (d_{ij} - d_{0j}) \quad \forall k$$

CONSTRUYENDO EL GRAFO

La lista de vecinos de cada nodo, excepto cero, se ordena ascendentemente según el peso de sus aristas. Para cero, se ordena descendenteamente.

Pero... ¿qué son α , β y μ ?

CONSTRUYENDO EL GRAFO

a representa el peso que se le va a dar a la distancia entre el nodo i y el nodo j a la hora de elegir el siguiente nodo que se añada a la ruta. $(1-a)$ representa el peso que se le dará a qué tan pronto se abre la ventana de tiempo del nodo j.

CONSTRUYENDO EL GRAFO

μ , por otra parte, trata de darle más peso a la distancia entre i y j según que tan lejos se esté del nodo 0.

Mientras más lejos se esté del 0 y menor sea d_{ij} , más se le restará al peso de la arista, por lo que el nodo j será priorizado.

CONSTRUYENDO EL GRAFO

¿Por qué β en vez de α para el nodo 0 ?

Esto se hace teniendo en cuenta que el nodo desde el que parte una ruta es muy importante, y queremos que al inicio se prioricen los nodos que estén lejos del 0 y tengan cierres muy tempranos.

CONSTRUYENDO EL GRAFO

Una última penalización...

A los vecinos del nodo 0 con sólo un vecino se les resta infinito al peso de su arista, pues se considera más eficiente que sean el último nodo de alguna ruta que contenga más nodos a que sean el único en nodo en la ruta (además del 0).

CONSTRUYENDO LAS RUTAS

Una vez construido el grafo, se construyen las rutas eligiendo siempre al primer vecino factible con la ruta, de acuerdo al orden de la lista de vecinos del nodo actual i .

ALGORITMO CONSTRUCTIVO CON RUVIDO



CONSTRUYENDO EL GRAFO

Se construye el grafo igual que en el método anterior. El único cambio es que ahora μ se usa para añadir ruido al peso de las aristas en lugar de como un coeficiente determinístico.

CONSTRUYENDO EL GRAFO

A la arista entre el nodo i y el nodo j se le asigna un peso de acuerdo al criterio:

$$\alpha \cdot d_{ij} + (1 - \alpha) \cdot e_j + ruido_{ij} \quad \forall i \neq 0$$

$$\beta \cdot d_{ij} - (1 - \beta) \cdot l_j + ruido_{ij} \quad i = 0$$

AÑADIENDO RUIDO

$ruido_{ij}$ se genera como un número de la distribución uniforme con parámetros:

$$U(0, \mu), \text{ si } \mu > 0$$

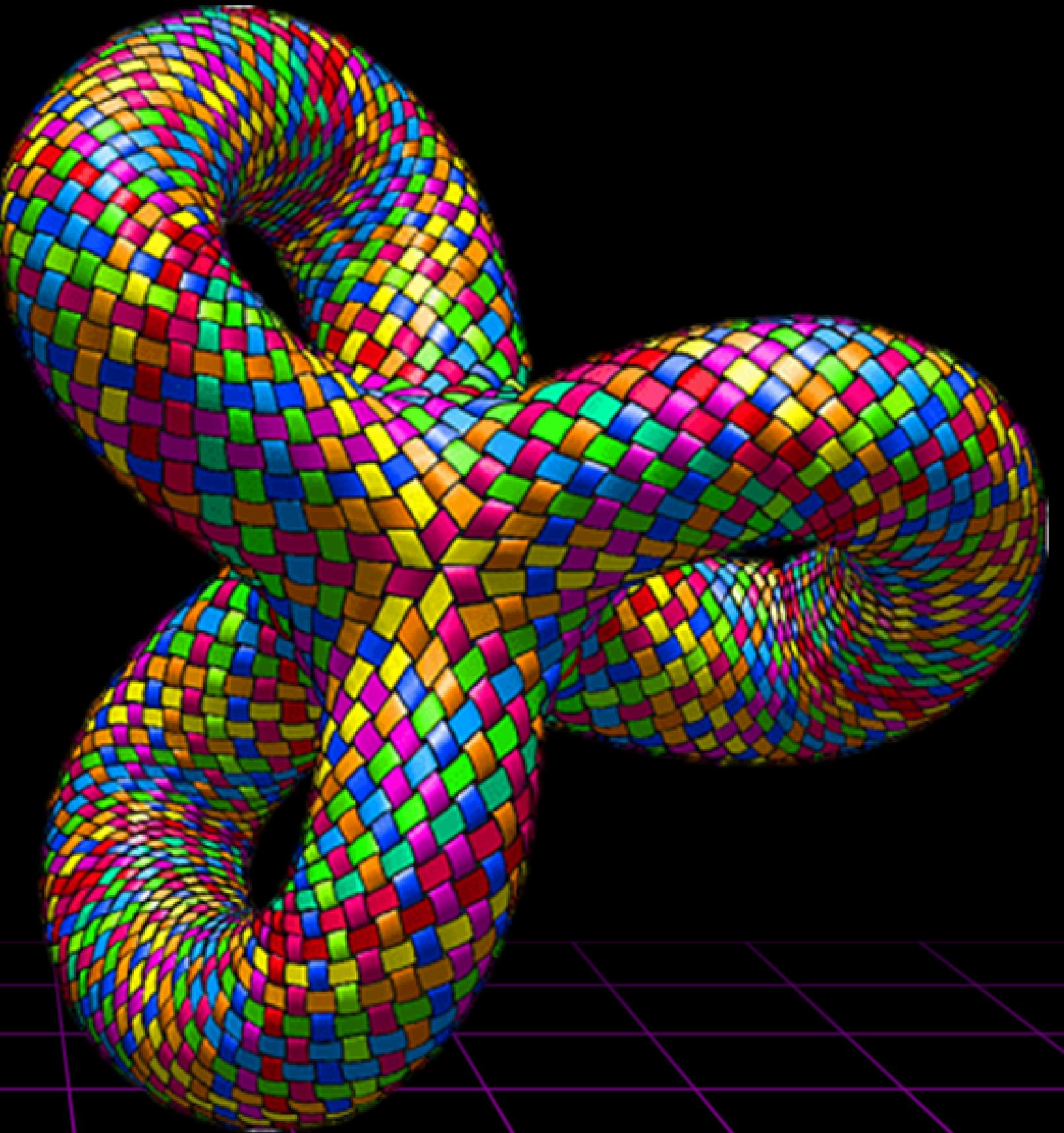
$$U(\mu, 0), \text{ si } \mu < 0$$

$$\mu = \frac{d_{0i}}{\max(d_{0k})} \cdot \frac{d_{0j}}{\max(d_{0k})} \cdot (d_{ij} - d_{0j})$$

CONSTRUYENDO LAS RUTAS

Una vez construido el grafo y añadido el ruido al peso de las aristas, se construyen las rutas igualmente que en el método anterior.

ALGORITMO GRASP



CONSTRUYENDO EL GRAFO

Se construye el grafo igual que en el primer método. El único cambio es que ahora μ también se usa para las aristas del nodo 0.

CONSTRUYENDO EL GRAFO

A la arista entre el nodo i y el nodo j se le asigna un peso de acuerdo al criterio:

$$\alpha \cdot d_{ij} + (1 - \alpha) \cdot e_j + \mu \quad \forall i \neq 0$$

$$\beta \cdot d_{ij} - (1 - \beta) \cdot l_j + \mu \quad i = 0$$

$$\mu = \frac{d_{0i}}{\max(d_{0k})} \cdot \frac{d_{0j}}{\max(d_{0k})} \cdot (d_{ij} - d_{0j})$$

CONSTRUYENDO LAS RUTAS

Una vez construido el grafo, se construyen las rutas.

En el primer y segundo método, elegíamos siempre al primer vecino factible con la ruta, de acuerdo al orden de la lista de vecinos del que fuera el nodo actual.

CONSTRUYENDO LAS RUTAS

Ahora, para el nodo actual i , construiremos una lista con todos sus vecinos factibles con la ruta y elegiremos uno de ellos aleatoriamente.

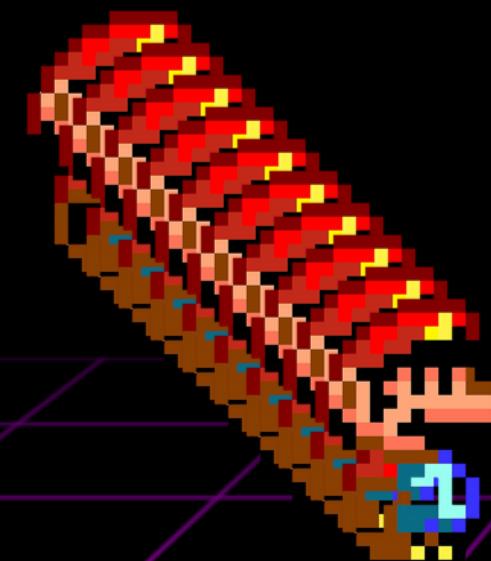
CONSTRUYENDO LAS RUTAS

La elección se realiza generando un número de la distribución geométrica con parámetro p . Si el número generado es mayor que el tamaño de la lista de candidatos, se genera otro número.

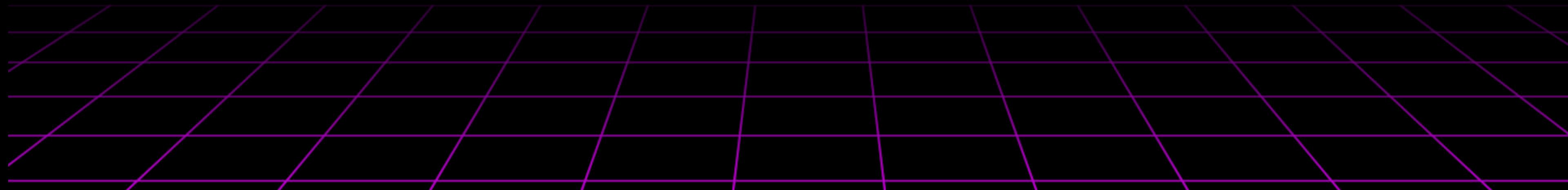
CONSTRUYENDO LAS RUTAS

Se seleccionó la distribución geométrica pues es discreta, por lo que sirve para generar el índice a elegir de la lista de candidatos, y está sesgada hacia la izquierda, con lo que tiende a elegir los mejores candidatos de acuerdo al peso de la arista.

RESULTADOS



COMPARACIÓN COTA INFERIOR



CÓMO SE OBTUVO LA COTA INFERIOR

Dado que fue muy (MUY) difícil hallar una cota inferior para el número de vehículos, se decidió hallar una cota inferior para la distancia total construyendo el árbol de mínima expansión teniendo en cuenta solo las coordenadas de los nodos.

QUÉ SE PUEDE DECIR

Esta cota parece estar considerablemente lejos del mínimo, además de que no es conveniente teniendo en cuenta que los algoritmos fueron hechos para primariamente minimizar número de vehículos, no distancia.

POR QUÉ NO ES MUY ÚTIL

En vista de esto, no es muy útil. Sin embargo, para el primer método el GAP promedio fue del 162.16%, para el segundo fue del 161.409% y para el tercero fue del 146.782%, para los mejores resultados hallados para cada método.

En vista de esto, se puede decir que los cambios introducidos para el método 1 fueron útiles para encontrar soluciones con menor distancia total.

ALGORITMO 1

INSTANCIA	alpha	beta	NÚMERO VEHÍCULOS	DISTANCIA TOTAL	COTA INFERIOR DISTANCIA	GAP (%)
1	0.5	0.5	3	191.82	98.990	93.772
7	0.91	0.7	5	363.25	195.660	85.653
13	0.5	0.5	10	838.76	417.300	100.998
2	0.5	0.5	2	215.54	141.823	51.980
8	0.5	0.5	2	497.17	264.105	88.249
14	0.5	0.5	3	591.56	492.470	20.120

ALGORITMO 1

INSTANCIA	alpha	beta	NÚMERO VEHÍCULOS	DISTANCIA TOTAL	COTA INFERIOR DISTANCIA	GAP (%)
3	0.6	0.4	8	660.16	276.778	138.514
9	0.6	0.4	12	1133.41	417.382	171.552
15	0.65	-1	21	1953.76	562.254	247.487
4	0.65	0.35	2	719.56	276.778	159.978
10	0.75	0.25	3	1179.64	417.382	182.628
16	0.87	-1	5	1599.72	562.254	184.519

ALGORITMO 1

INSTANCIA	alpha	beta	NÚMERO VEHÍCULOS	DISTANCIA TOTAL	COTA INFERIOR DISTANCIA	GAP (%)
5	0.2	0.2	4	488.45	178.406	173.783
11	0.45	-1	9	1059.97	315.738	235.712
17	0.7	0.3	17	1879.78	563.998	233.296
6	0.7	0.35	2	572.39	178.406	220.833
12	0.65	0.1	3	1333.70	315.738	322.407
18	0.84	0	5	1707.24	563.998	202.703

ALGORITMO 2

INSTANCIA	alpha	beta	NÚMERO VEHÍCULOS	DISTANCIA TOTAL	COTA INFERIOR DISTANCIA	GAP (%)
1	0.5	0.5	3	191.815	98.99	93.772
7	0.91	0.7	5	407.026	195.66	108.027
13	0.5	0.5	10	946.856	417.3	126.901
2	0.5	0.5	2	328.895	141.823	131.905
8	0.5	0.5	2	497.174	264.105	88.249
14	0.5	0.5	3	762.052	492.47	54.741

ALGORITMO 2

INSTANCIA	alpha	beta	NÚMERO VEHÍCULOS	DISTANCIA TOTAL	COTA INFERIOR DISTANCIA	GAP (%)
3	0.6	0.4	8	660.155	276.778	138.514
9	0.6	0.4	12	1124.37	417.382	169.386
15	0.65	-1	20	1815.71	562.254	222.934
4	0.65	0.35	2	647.436	276.778	133.919
10	0.75	0.25	3	1150.46	417.382	175.637
16	0.87	-1	5	1533.43	562.254	172.729

ALGORITMO 2

INSTANCIA	alpha	beta	NÚMERO VEHÍCULOS	DISTANCIA TOTAL	COTA INFERIOR DISTANCIA	GAP (%)
5	0.2	0.2	4	470.616	178.406	163.789
11	0.45	-1	8	988.526	315.738	213.084
17	0.7	0.3	16	1876.06	563.998	232.636
6	0.7	0.35	2	531.343	178.406	197.828
12	0.65	0.1	3	1199.92	315.738	280.037
18	0.84	0	4	1699.14	563.998	201.267

ALGORITMO 3

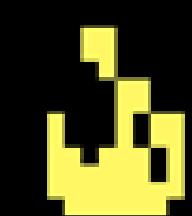
INSTANCIA	alpha	beta	p	NÚMERO VEHÍCULOS	DISTANCIA TOTAL	COTA INFERIOR DISTANCIA	GAP (%)
1	0.5	0.5	0.95	3	191.815	98.99	93.772
7	0.91	0.7	0.95	5	370.034	195.66	89.121
13	0.5	0.5	0.95	10	907.854	417.3	117.554
2	0.5	0.5	0.95	2	275.139	141.823	94.002
8	0.5	0.5	0.95	2	444.959	264.105	68.478
14	0.5	0.5	0.95	3	603.878	492.47	22.622

ALGORITMO 3

INSTANCIA	alpha	beta	p	NÚMERO VEHÍCULOS	DISTANCIA TOTAL	COTA INFERIOR DISTANCIA	GAP (%)
3	0.6	0.4	0.95	8	630.379	276.778	127.756
9	0.6	0.4	0.95	12	1088.08	417.382	160.692
15	0.65	-1	0.95	20	1880.76	562.254	234.504
4	0.65	0.35	0.95	2	577.368	276.778	108.603
10	0.75	0.25	0.95	3	1065.58	417.382	155.301
16	0.87	-1	0.95	4	1696.01	562.254	201.645

ALGORITMO 3

INSTANCIA	alpha	beta	p	NÚMERO VEHÍCULOS	DISTANCIA TOTAL	COTA INFERIOR DISTANCIA	GAP (%)
5	0.2	0.2	0.95	4	470.951	178.406	163.977
11	0.45	-1	0.95	9	993.748	315.738	214.738
17	0.7	0.3	0.95	16	1805.32	563.998	220.093
6	0.7	0.35	0.95	2	461.359	178.406	158.601
12	0.65	0.1	0.95	3	1041.56	315.738	229.881
18	0.84	0	0.95	5	1583.35	563.998	180.737



COMPARACIÓN ENTRE LOS MÉTODOS (FINALMENTE)

LO QUE SE PUO

El número de vehículos promedio del método 1 fue de 6.44, del 2 fue de 6.22 y del 3 fue de 6.27. Las distancias promedios fueron de 944.56, 911.75 y 886.48, respectivamente. A continuación se presentan los resultados de cada método para cada instancia y se determina ganador para cada una.

COMPETENCIA

INSTANCIA	N. VEHÍCULOS 1	DISTANCIA 1	N. VEHÍCULOS 2	DISTANCIA 2	N. VEHÍCULOS 3	DISTANCIA 3	WINNER
1	3	191.815	3	191.815	3	191.815	EMPATE
7	5	363.248	5	363.248	5	363.248	EMPATE
13	10	855.065	10	855.065	10	855.065	EMPATE
2	2	215.542	2	215.542	2	215.542	EMPATE
8	2	497.174	2	497.174	2	444.959	3
14	3	591.555	3	591.555	3	591.555	EMPATE

COMPETENCIA

INSTANC IA	N. VEHÍCULOS 1	DISTANC IA 1	N. VEHÍCUL OS 2	DISTANC IA 2	N. VEHÍCUL OS 3	DISTANCIA 3	WINNER
3	8	660.155	8	660.155	8	630.379	3
9	12	1133.41	12	1124.37	12	1088.08	3
15	21	1953.76	20	1815.71	20	1880.76	2
4	2	719.561	2	647.436	2	577.368	3
10	3	1179.64	3	1150.46	3	1065.58	3
16	5	1599.72	5	1533.43	4	1696.01	3

COMPETENCIA

INSTANC IA	N. VEHÍCULOS 1	DISTANC IA 1	N. VEHÍCUL OS 2	DISTANC IA 2	N. VEHÍCUL OS 3	DISTANCIA 3	WINNER
5	4	488.446	4	470.616	4	470.951	2
11	9	1059.97	8	988.526	9	993.748	2
17	17	1879.78	16	1876.06	16	1805.32	3
6	2	572.386	2	531.343	2	461.359	3
12	3	1333.7	3	1199.92	3	1041.56	3
18	5	1707.24	4	1699.14	5	1583.35	2

IL CAMPEONE

Vemos que el método 3 ganó 7 desempates por distancia y 1 por número de vehículos, mientras que el método 2 ganó 2 desempates por distancia y 2 por número de vehículos, por lo cual es difícil decidir un ganador. Sin embargo, gana el 2 porque lo que nos importa es número de vehículos.

BONO (TIEMPOS DE COMPUTO)

Los tiempos promedios de ejecución en milisegundos fueron de 0 para el primer método, 1.722 para el segundo y 1.33 para el tercero. Como ningún método realiza búsqueda local y los tres siguen esencialmente el mismo proceso para la construcción del grafo y la ruta, no sorprende que la diferencia sea despreciable.

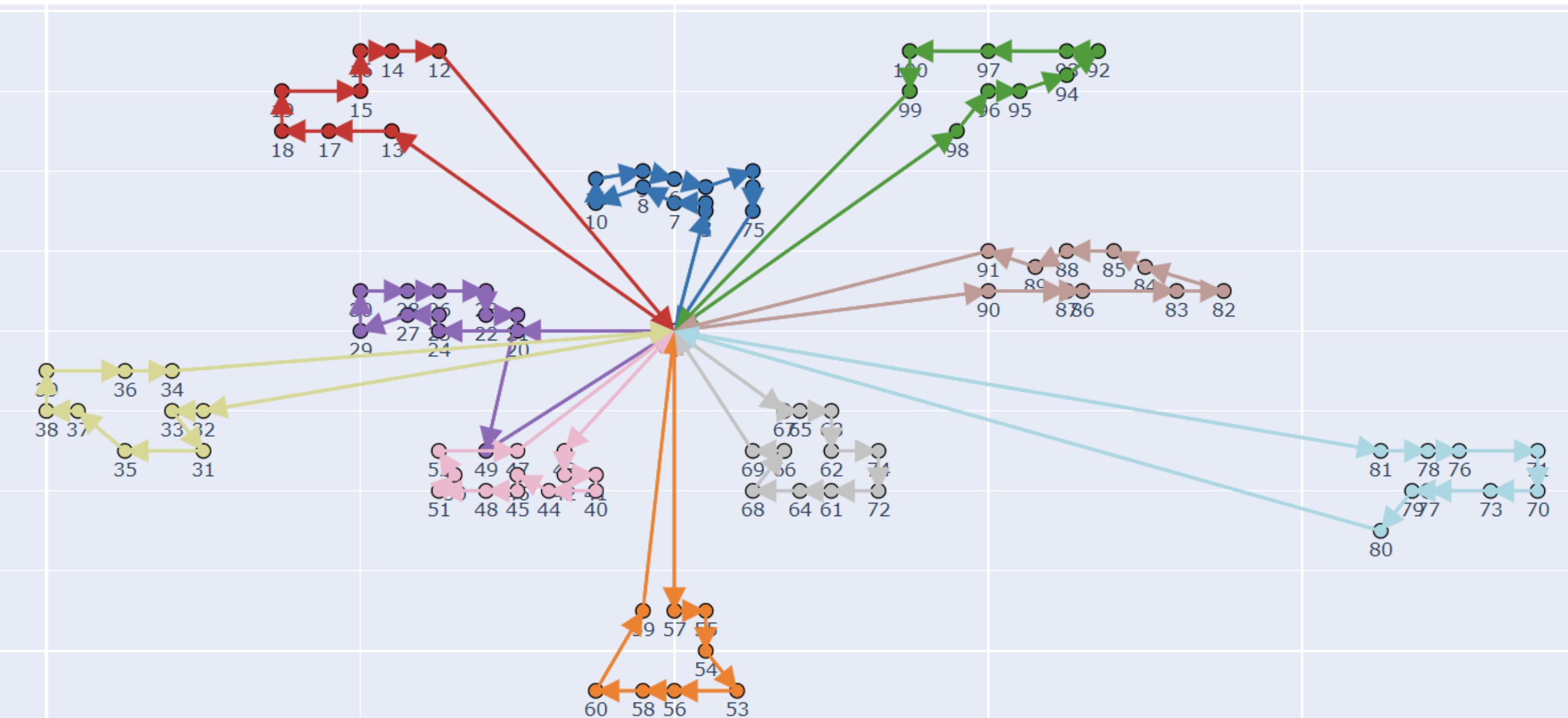
COMPARACIÓN DE PARÁMETROS

A VER

A continuación veremos las rutas creadas para una instancia por diferentes parametrizaciones del mismo método con el fin de apreciar la diferencia que hace que el alpha, el beta o el p se cambien.

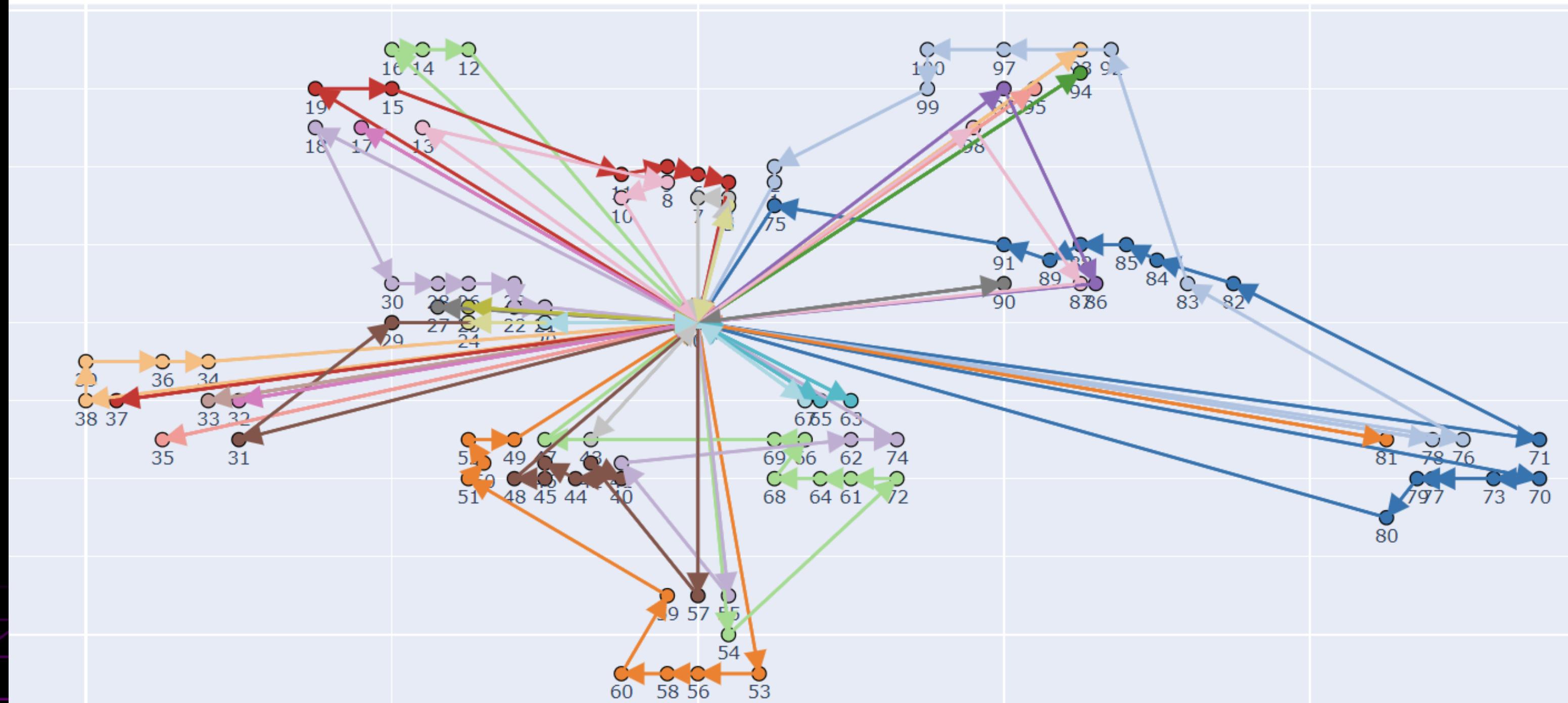
INSTANCIA 13

método 1: alpha=0.5, beta=0.5



INSTANCIA 13

método 1: alpha=0, beta=1



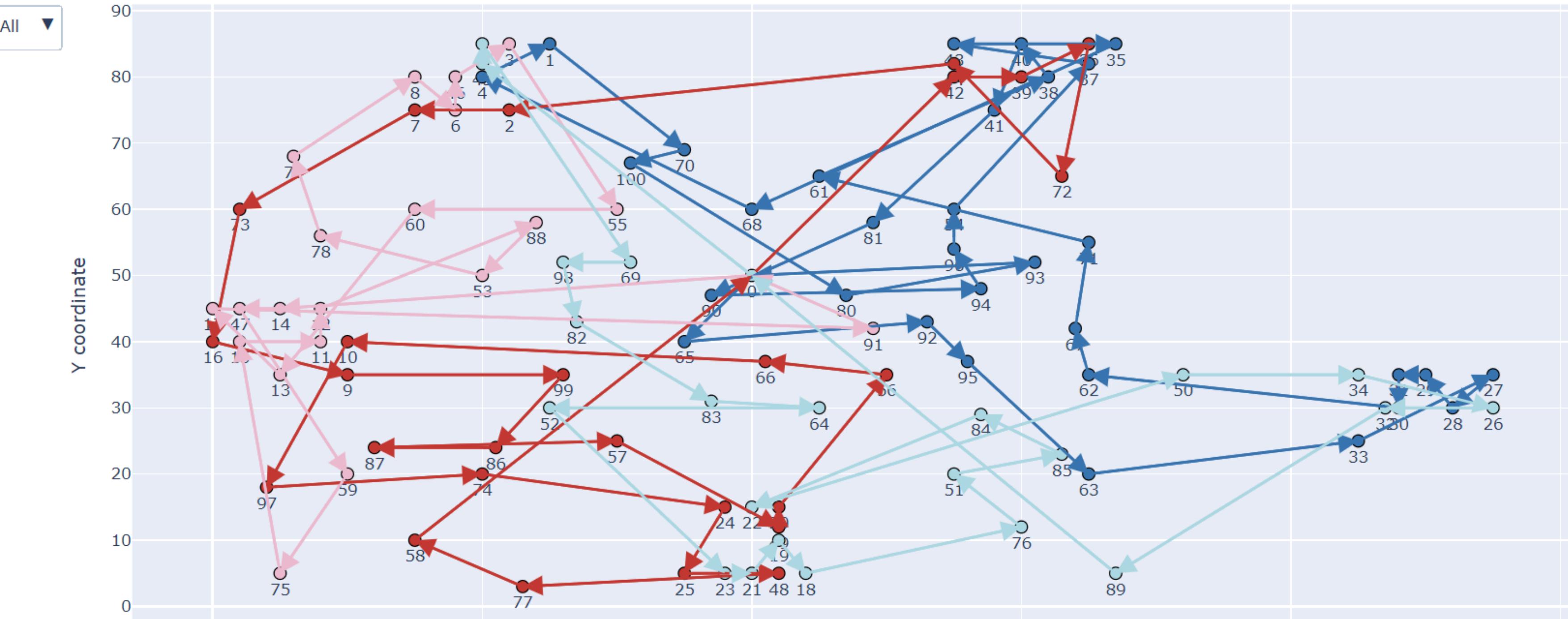
QUÉ SE VE (Y UN POCO MÁS)

Las dos imágenes anteriores presentan las rutas producidas por el método para el método 13, la mejor encontrada junto con una intencionalmente mala para resaltar la importancia de elegir parámetros acorde a la naturaleza del problema.

Como estos nodos se encuentran clusterizados y las ventanas de tiempo no son considerablemente amplias, debe tenerse en cuenta tanto la distancia como las ventanas de tiempo para lograr una solución buena. El $\alpha=0$ hace que solo se guíe con base en la prontitud de las aperturas de la ventana de tiempo.

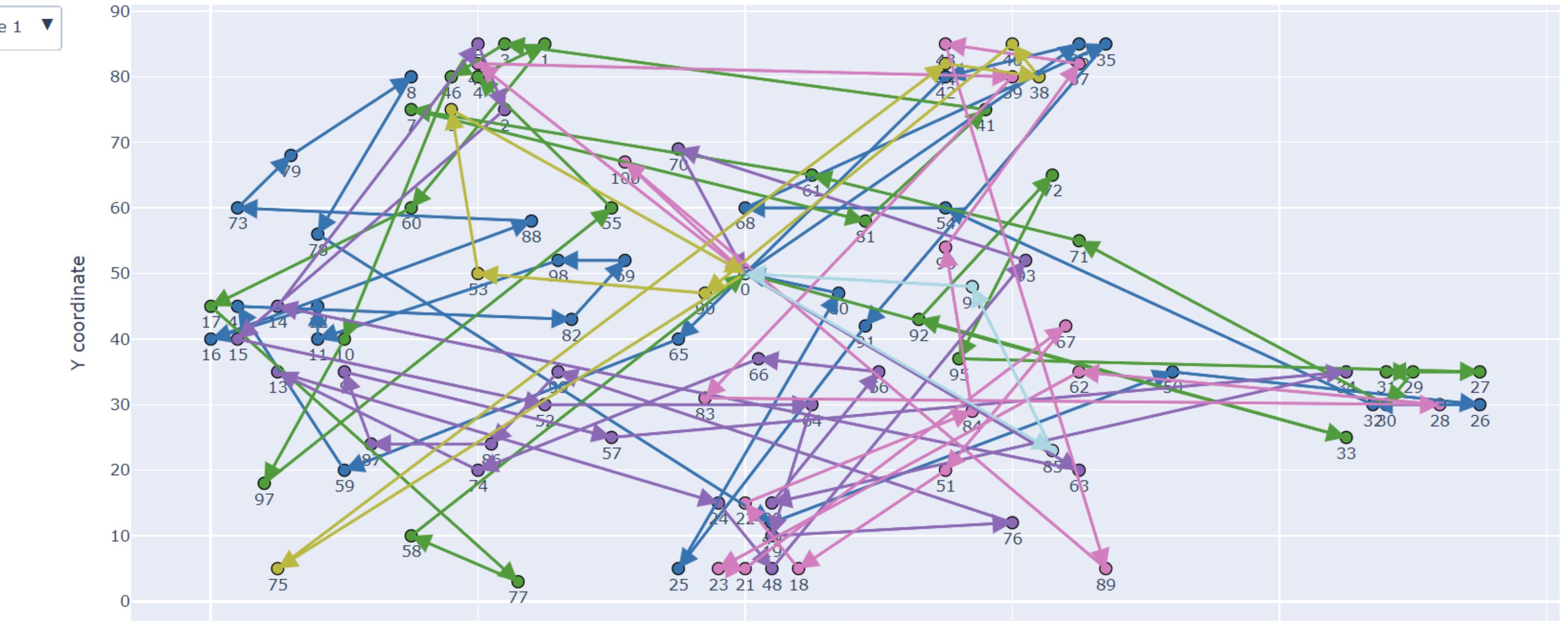
INSTANCIA 18

Instancia 18, método 2: alpha=0.84, beta=0, distancia=1699.14 , nvehiculos=4



INSTANCIA 18

Instancia 18, método 2: alpha=0.3, beta=0.8, distancia=3054.54, nvehiculos=6

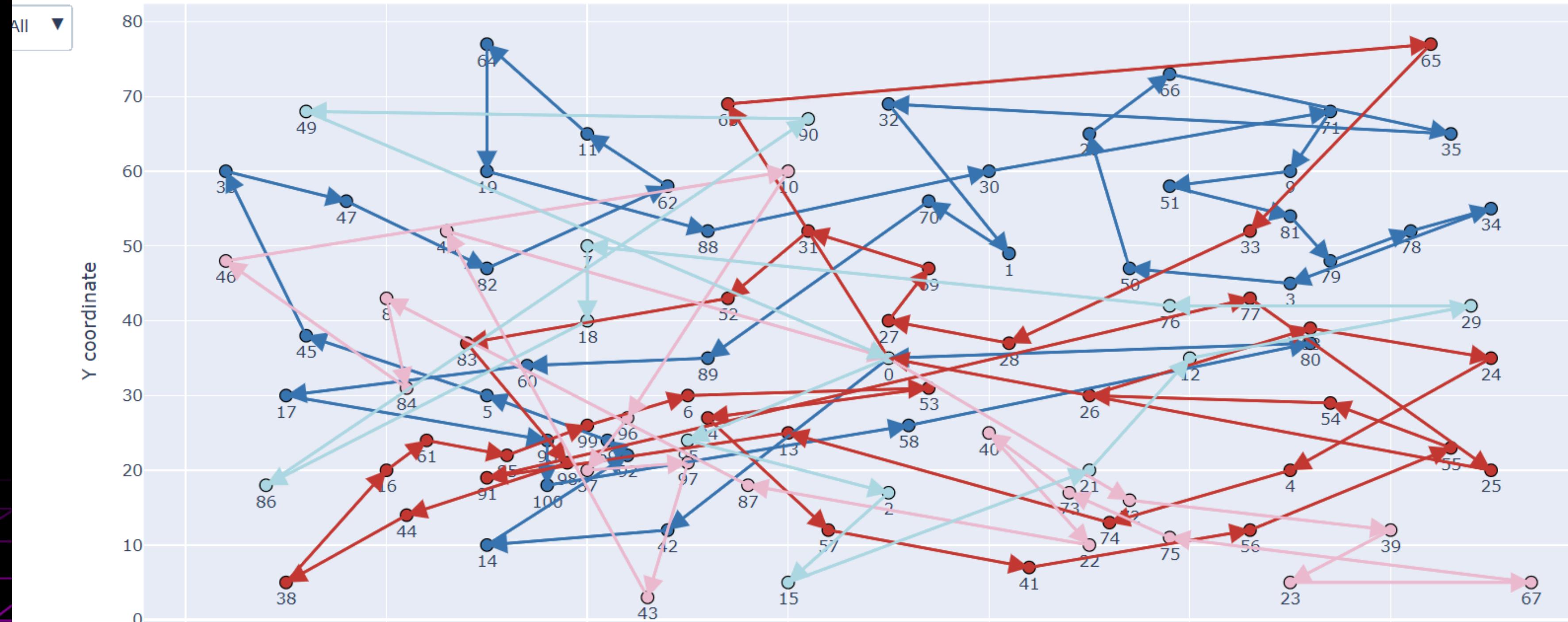


QUÉ SE VE (Y UN POCO MÁS)

Estos nodos parecen estar un poco clusterizados y un poco aleatorizados y las ventanas de tiempo de todos son iguales (120). Por esoto la distancia juega un rol muy importante, ya que nos permite recorrer los clusteres en menos rutas si le damos más peso en los coeficientes. Es por esto que si bien el método logra un número decente de rutas priorizando las ventanas de tiempo (6 vehículos y $\alpha=0.3$), darle más peso a la distancia logra mejores resultados.

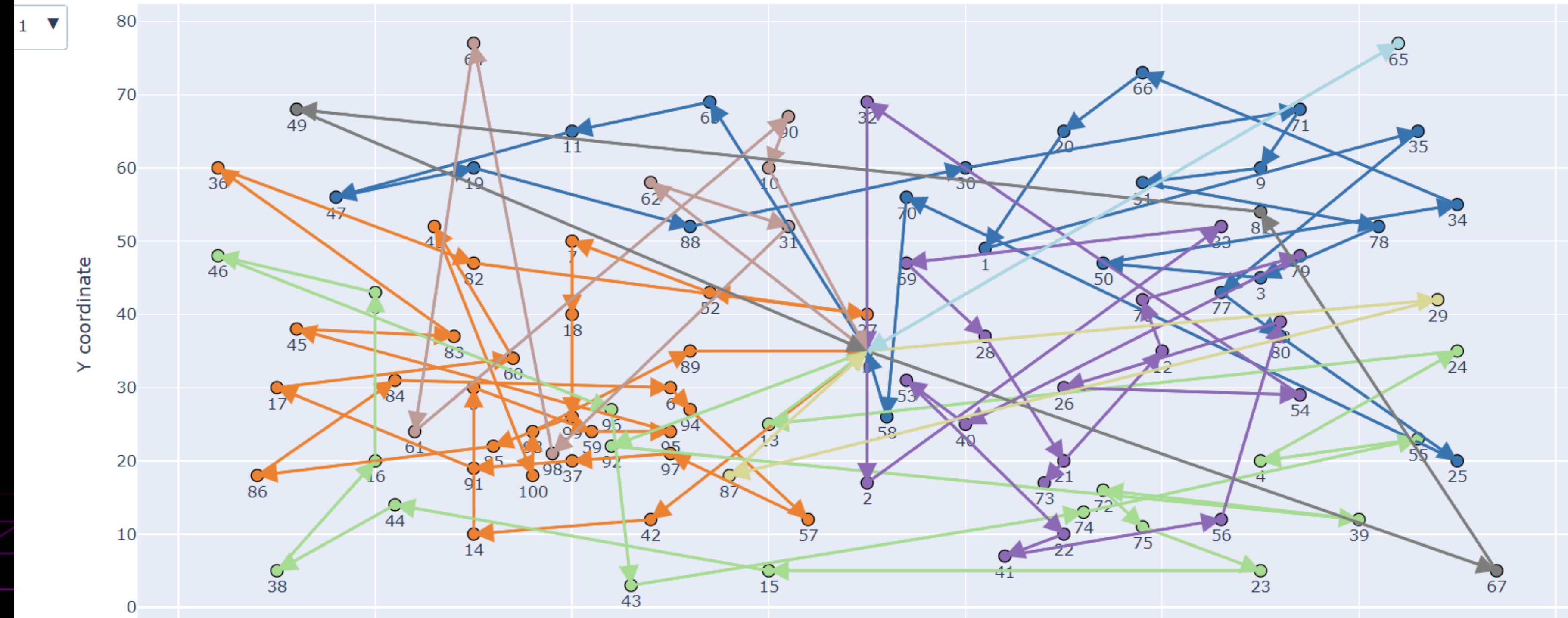
INSTANCIA 16

Instancia 16, método 3: alpha=0.87, beta=-1, p=0.95, distancia=1696.01, nvehiculos=4



INSTANCIA 16

Instancia 16, método 3: alpha=0.87, beta=-1, p=0.5, distancia=2143.26, nvehiculos=8



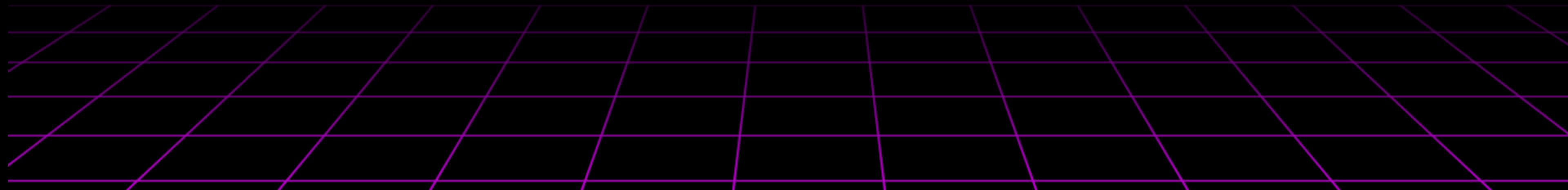
QUÉ SE VE (Y UN POCO MÁS)

Esta instancia en particular era bastante desafiante porque la ubicación de los nodos parecía completamente aleatoria y las ventanas de tiempo variaban bastante (había nodos con ventanas de tiempo amplias y otros con ventanas de tiempo cortas).

Lo que se puede apreciar en las dos soluciones anteriores es que parece convenir sesgar la elección de candidatos hacia las mejores (que p sea más alto), pues a pesar de que alpha y beta eran iguales, lograron soluciones muy disímiles.



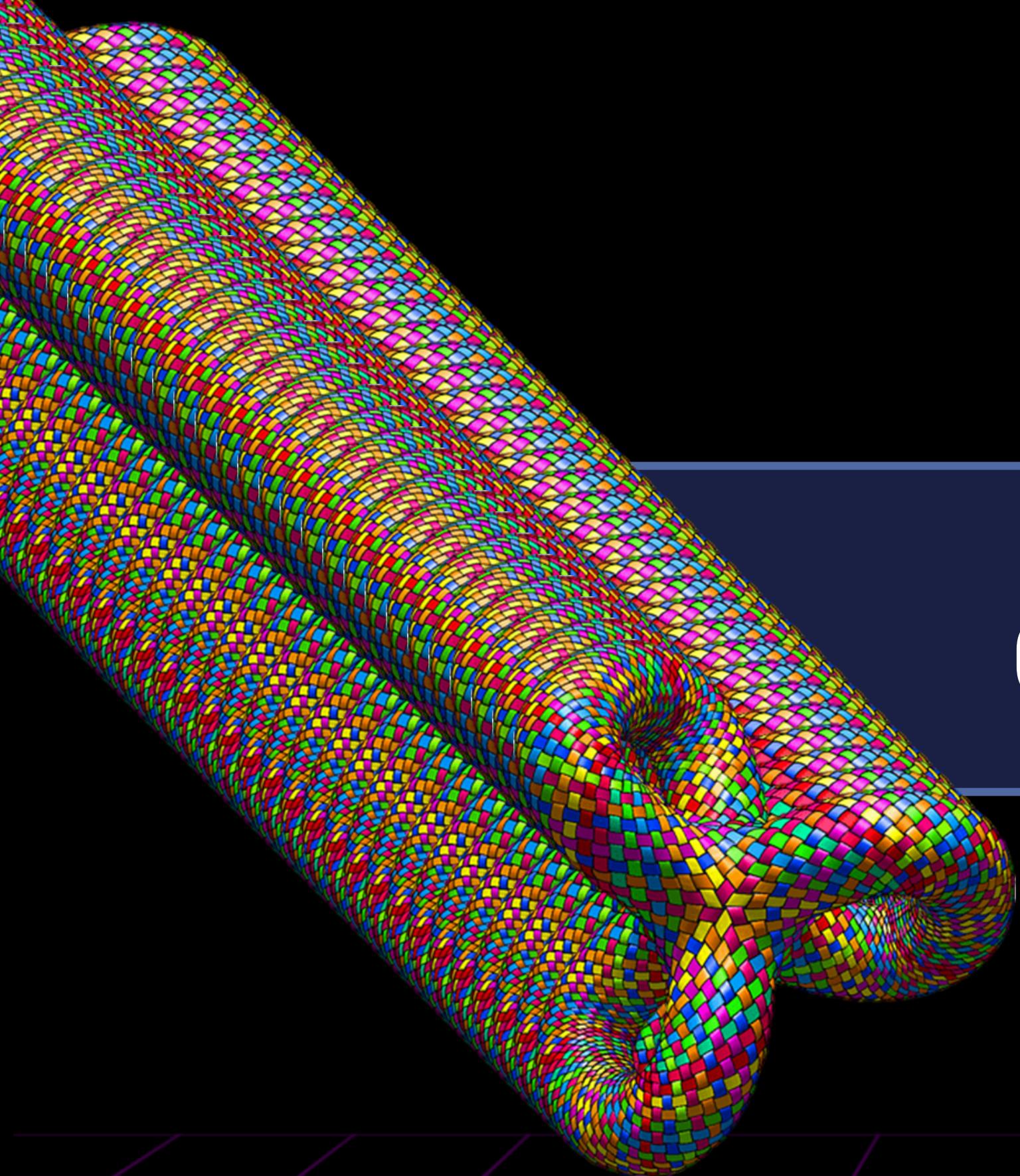
CONCLUSIONES



UMMM

Lo que puedo concluir después de hacer este taller es que la aleatoriedad es útil para lograr soluciones más diversas, pero sobre todo es útil mientras mayor sea la dimensinalidad del problema.

También me sorprendió que un algoritmo constructivo que es completamente determinista generara soluciones aceptables solo teniendo en cuenta características “locales” del problema y me gustó introducir ruido y aleatoriedad sesgando en función de lo que me parecía más conveniente para construir mejores soluciones.



GRACIAS!

EL FIN