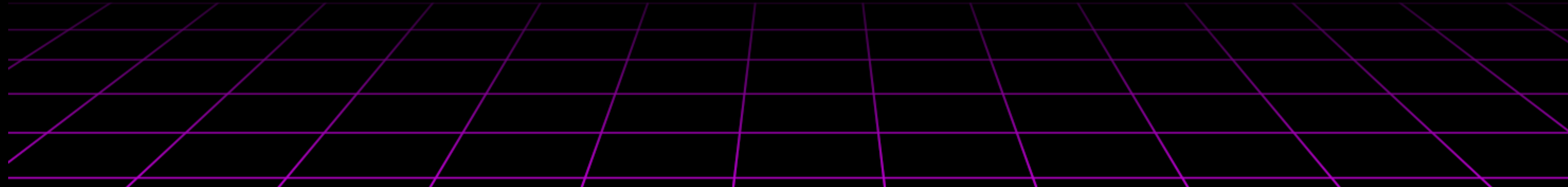


ALGORITMOS EVOLUTIVOS

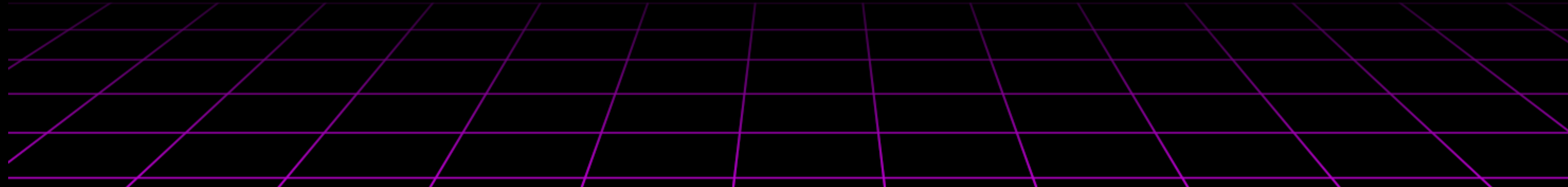
CAMILO
BERMÚDEZ

GENETIC ALGORITHM (GA)



REPRESENTACIÓN DEL CROMOSOMA

Representamos un cromosoma como una cadena de tamaño n , donde n es el número de clientes. Esto representa una secuencia de rutas, aunque no se usa ningún delimitador para indicar el inicio o el fin de una ruta.



GENERACIÓN SOLUCIONES INICIALES

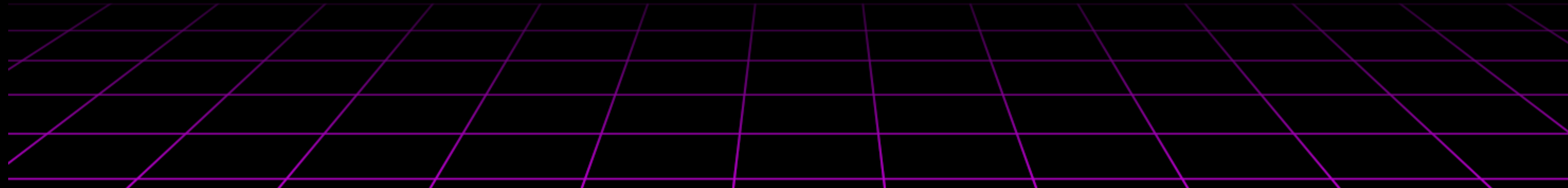
k ($0 \leq k \leq 100\%$) de las soluciones iniciales son generadas aleatoriamente. Es decir, se permuta una cada con todos los números entre 1 y n . El resto de la soluciones son generadas con un algoritmo greedy que prioriza la distancia, añadiendo siempre el cliente más cercano dentro de cierto radio.

DECODING

Nuestro proceso de decoding consiste de dos fases. En la primera se transforma el cromosoma a una solución válida. Un vehículo parte del depósito y el primer gen del cromosoma indica el primer cliente a visitar. Se siguen añadiendo clientes en el orden que aparecen mientras sea posible, y se añade una nueva ruta cada vez que se encuentra un cliente que no puede ser añadido a la ruta actual.

DECODING

En la segunda fase, el último cliente de cada ruta se trata de reinsertar como el primer cliente de la siguiente, y en caso de que se respeten las restricciones de tiempo y de carga y se mejore la función objetivo, la reinsertación es realizada.

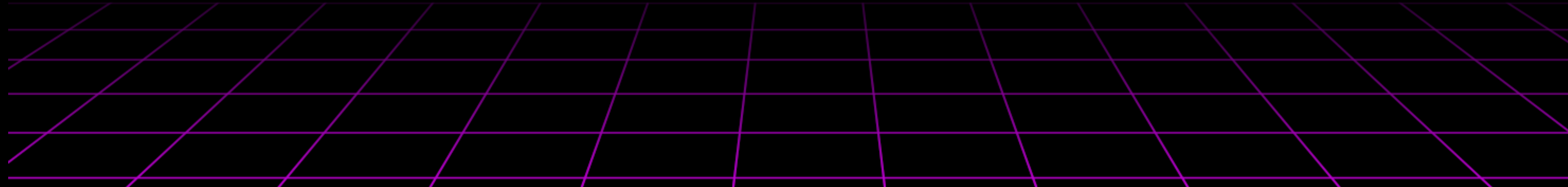


PARETO RANKING

Dominancia de Pareto: Una solución A domina a otra solución B si A es mejor o igual que B en todos los objetivos y A es estrictamente mejor que B en al menos un objetivo. Las soluciones se clasifican en rangos: Rango 1: No dominadas por ninguna otra solución. Rango 2: Dominadas solo por soluciones de Rango 1. Rango 3: Dominadas por Rango 1 y Rango 2, y así sucesivamente.

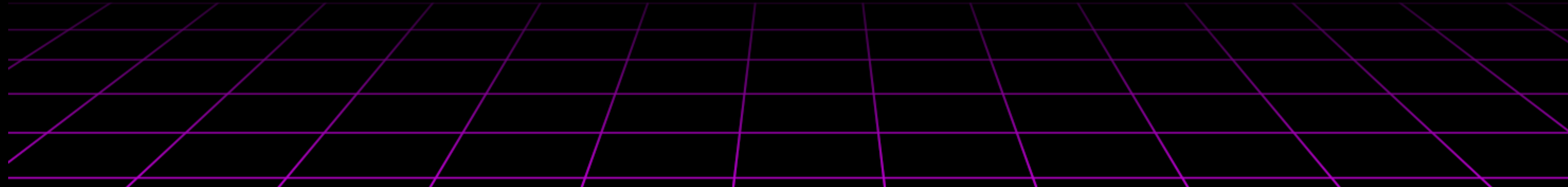
PARETO RANKING

El esquema de Pareto ranking reemplaza la función de fitness tradicional, y a todas las soluciones de la población se les es asignado un rango de Pareto de acuerdo a lo descrito en la diapositiva anterior.



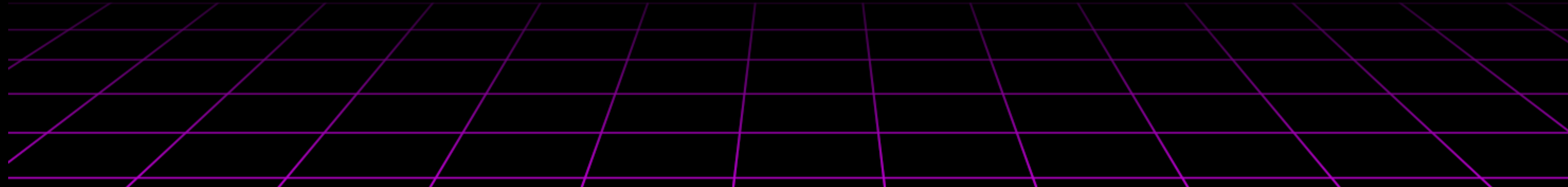
FITNESS-BASED SELECTION

Mantenemos el mismo tamaño de población en todas las generaciones. Todas las soluciones con rango 1 son mantenidas en la siguiente generación y el resto son generadas emparejando soluciones como se describe a continuación.



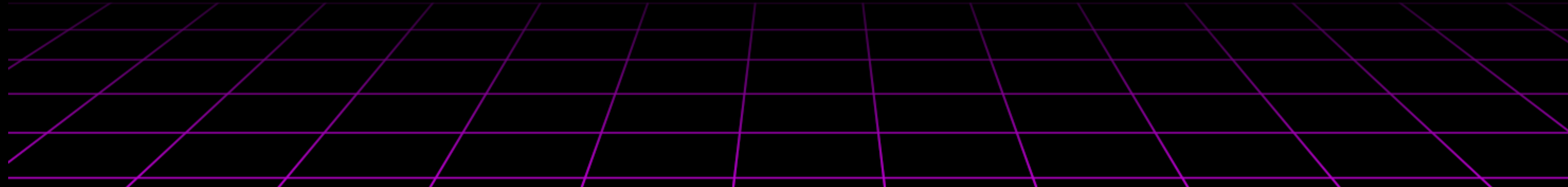
FITNESS-BASED SELECTION

En cada generación debemos seleccionar padres para emparejamiento y reproducción. La elección de un padre funciona del siguiente modo en cada etapa: Se eligen cuatro individuos aleatorios y se selecciona el mejor (aquel con menor rango de pareto) con probabilidad r , o uno aleatorio con probabilidad $1-r$.



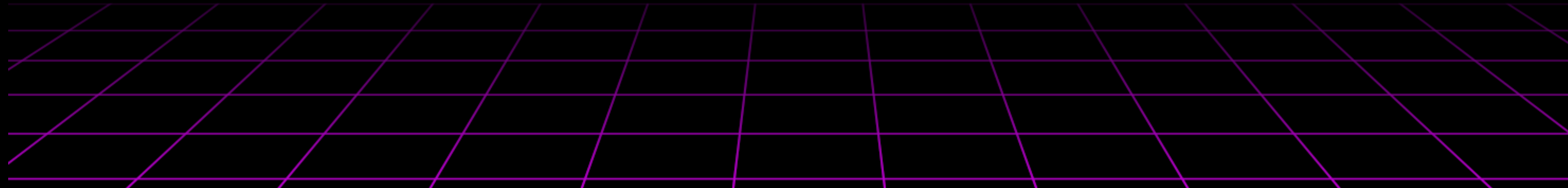
RECOMBINACIÓN

Una vez elegidos dos padres, se aplica el algoritmo de recombinación con probabilidad `CROSSOVER_RATE`, y en caso de que no sea aplicado, ambos padres son añadidos a la siguiente generación.



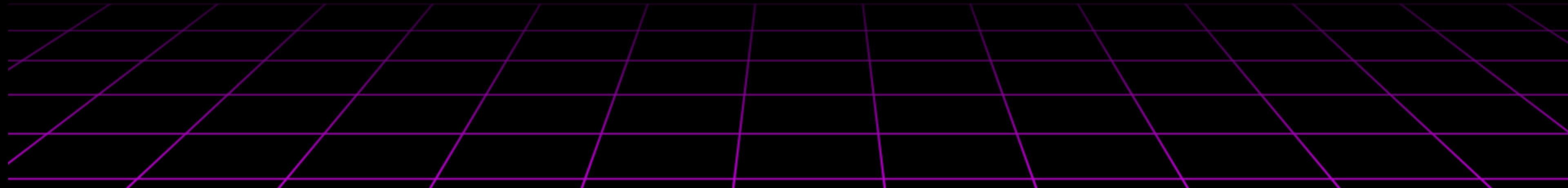
FASE DE RECOMBINACIÓN

De cada padre se selecciona una ruta aleatoria. Los clientes contenidos en estas rutas son eliminados en las rutas del otro padre y, dado que toda solución debería contener los n clientes, estos son reinsertados de modo que se minimice la función objetivo. Así son producidas dos nuevas soluciones.



FASE DE MUTACIÓN

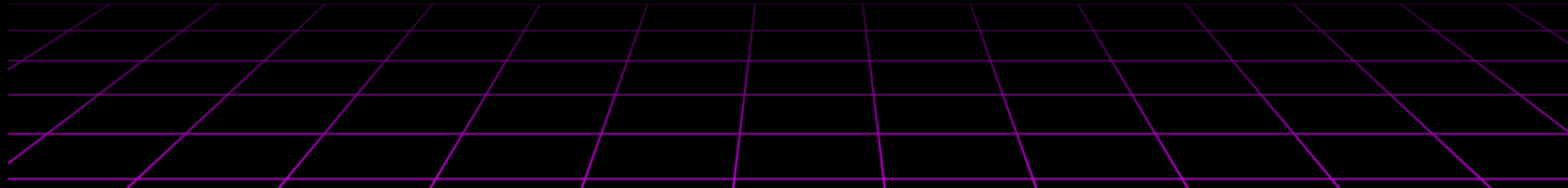
Para cada individuo de la nueva generación, se realice una mutación con probabilidad `MUTATION_RATE`, que consiste en reversar un segmento de dos o tres nodos dentro de la misma ruta de modo que se respeten las restricciones de tiempo y no se empeore la función objetivo.



ALGORITMO GA

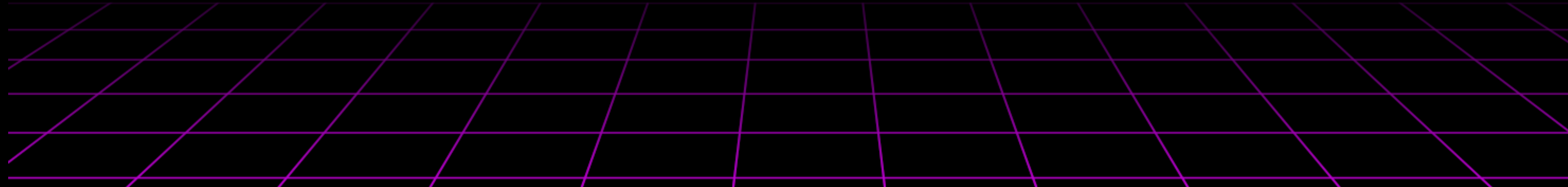
```
Generar soluciones_iniciales
poblacion_actual = soluciones_iniciales
for gen=1 to num_generations {
    decode(poblacion_actual)
    pareto_rank(poblacion_actual)
    select_new_population(poblacion_actual)
    apply_ga_operators(poblacion_actual)
    //crossover & mutation
}
```

GENETIC ALGORITHM WITH VNO (GA-VNO)



SOLO UNA COSA

Se aplica exactamente el mismo algoritmo evolutivo descrito anteriormente, y se aplica VND a la mejor solución de la última generación. El algoritmo de VND es el mismo usado en el trabajo anterior.



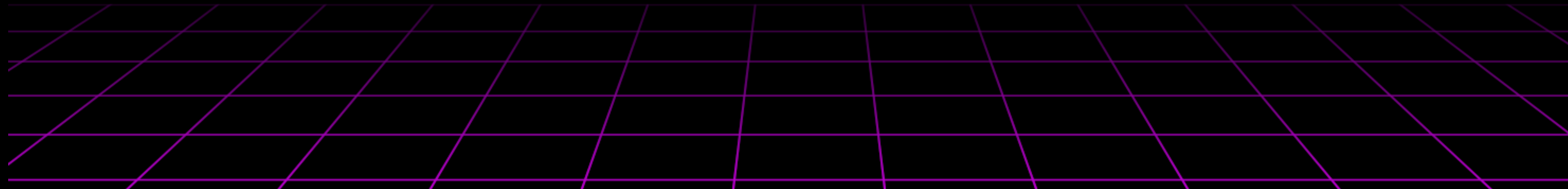
SOLO UNA COSA

Como vecindarios se usaron los operadores shift (saca un nodo de una ruta y lo trata de insertar en otra distinta), rearrrange (reordena un segmento dentro de la misma ruta), exchange (intercambia el segmento final entre dos rutas), SWAP (intercambia dos nodos entre rutas diferentes) y reverse (reversa el orden de un segmento dentro de la misma ruta).

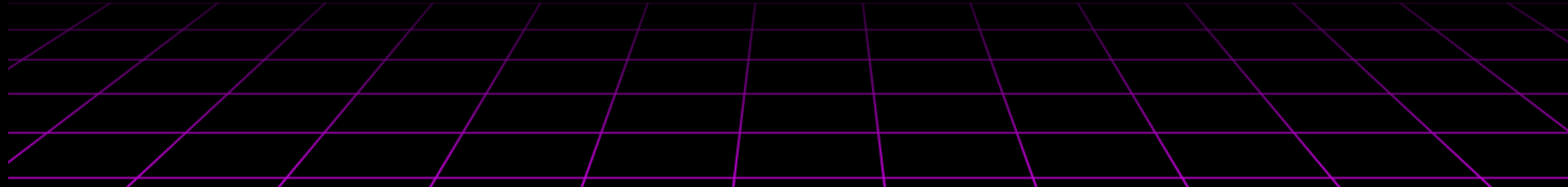
SOLO UNA COSA

Se decidió seguir el orden shift -> rearrange -> exchange -> SWAP -> reverse, pues intercala operadores que pueden reducir el número de rutas con operadores que pueden mejorar la distancia recorrida de la configuración actual. El resto del algoritmo siguió la implementación vista en clase para un VNS básico (estructura de VND con perturbación y búsqueda local).

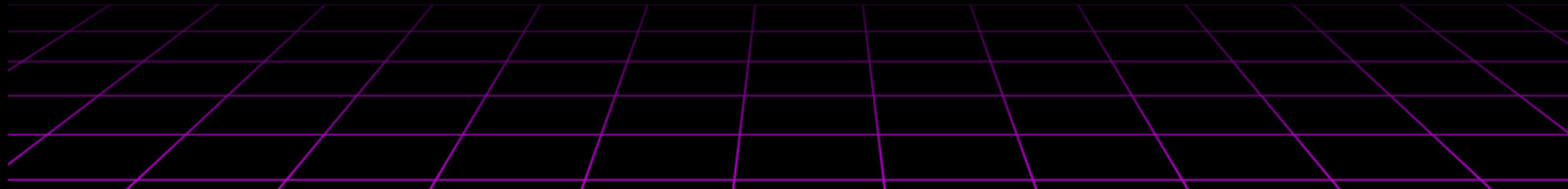
RESULTADOS



COMPARACIÓN MÉTODOS



COMPETENCIA NÚMERO DE VEHÍCULOS



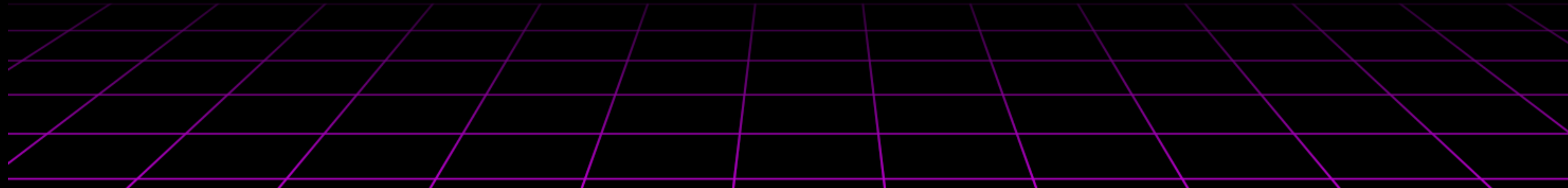
[illegible]

[illegible]

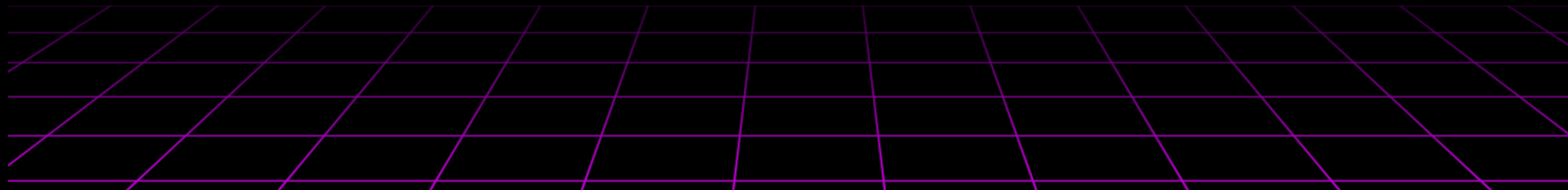
inst anc ia	GA	GA- VND	PAR ALLE L SA	PARALL EL VNS	CONSTR UCTIVO	RUIDO	GRASP	BEST SHIFT	FIRST SHIFT	OTHER LOCAL S
13	10	10	10	10	10	10	12	10	10	10
14	3	3	3	3	3	3	4	3	3	3
15	19	19	19	20	21	20	20	20	20	20
16	4	4	4	4	5	5	4	4	4	4
17	15	15	16	17	17	16	16	16	16	16
18	4	4	4	4	5	4	5	4	4	4

QUÉ SE PUEDE DECIR

Como se pudo apreciar, ambos algoritmos genéticos igualaron al parallel simulated annealing en todas las instancias, excepto en la instancia 17, en la que ambos lograron reducir el número de vehículos en 1. Con lo que ambos algoritmos genéticos son los claros vencedores.



COMPETENCIA GAP
(%) NÚMERO DE
VEHÍCULOS



[illegible]

[illegible]

inst anc ia	GA	GA- VN D	PARAL LEL SA	PARALL EL VNS	CONSTR UCTIVO	RUIDO	GRASP	BEST SHIFT	FIRST SHIFT	OTHER LOCALS
13	0	0	0	0	0	0	20	0	0	0
14	0	0	0	0	0	0	33.3	0	0	0
15	137 .5	137 .5	137.5	150	162.5	150	150	150	150	150
16	100	100	100	100	150	150	100	100	100	100
17	66. 66	66. 66	77.77 778	88.888 89	88.888 89	77.777 78	77.77 778	77.77 778	77.77 778	77.777 78
18	100	100	100	100	150	100	150	100	100	100

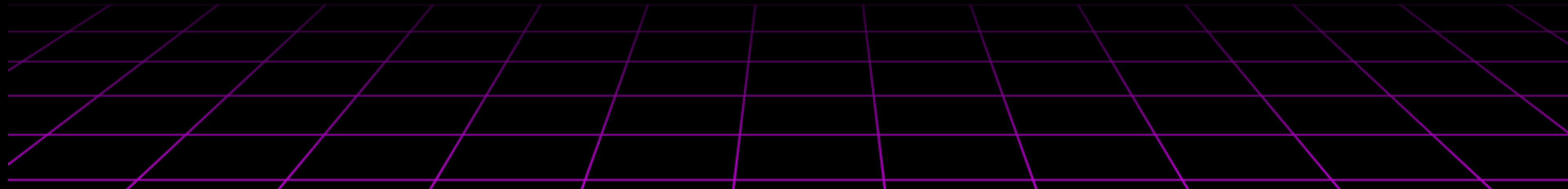
¿QUÉ SE PUEDE DECIR?

Con respecto a las soluciones logradas por los métodos presentados en los trabajos previos, el parallel simulated annealing es el que más parejo está, pues solo difiere en la instancia 17. Y esto se refleja en el GAP promedio alcanzado por ambos algoritmos genéticos, pues ambos lograron un GAP promedio de 87.37%, mientras el parallel había alcanzado un GAP promedio de 87.98%, con lo que ambos son claros vencedores. Una vez más!!

COMPETENCIA GAP

(%) DISTANCIA

TOTAL



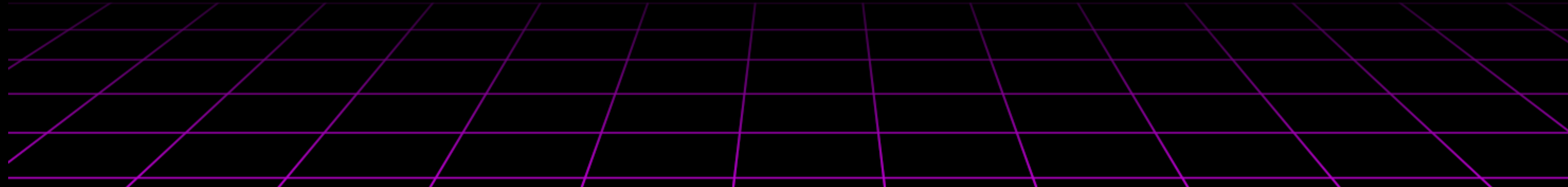
inst anci a	GA	GA- VND	PARAL LEL SA	PARAL LEL VNS	CONST RUCTI VO	RUIDO	GRASP	BEST SHIFT	FIRST SHIFT	OTHER LOCALS
1	43.91	43.91	43.91	43.91	43.91	43.91	43.91	43.91	43.91	43.91
2	23.89	17.02	17.02	17.02	17.02	17.02	58.96	17.02	17.02	17.02
3	56.27	54.90	54.69	54.69	65.15	65.15	57.70	54.90	54.90	57.70
4	66.52	66.52	66.82	66.82	128.8 2	105.8 8	83.60	77.50	73.88	83.60
5	184.0 3	184.03	184.0 3	193.13	200.1 9	189.2 3	189.4 4	188.19	188.19	189.23
6	129.17	129.17	128.15	128.15	202.0 8	180.4 2	143.4 9	143.49	143.4 9	143.49

inst anci a	GA	GA- VND	PARAL LEL SA	PARAL LEL VNS	CONST RUCTIV O	RUID O	GRASP	BEST SHIFT	FIRST SHIFT	OTHER LOCALS
7	43.92	43.92	43.92	43.92	43.92	43.92	74.61	43.92	43.92	43.92
8	45.40	45.40	45.40	45.40	62.47	62.47	45.40	45.40	45.40	45.40
9	77.74	77.74	76.44	67.78	81.68	80.23	74.41	69.07	68.80	74.41
10	89.35	91.02	98.69	69.94	129.99	124.30	107.75	91.06	95.36	107.75
11	215.67	215.67	217.43	219.80	253.53	229.70	231.44	224.31	224.05	229.70
12	234.18	232.42	238.15	238.15	428.72	375.69	312.91	312.91	312.91	312.91

inst anc ia	GA	GA- VND	PARAL LEL SA	PARAL LEL VNS	CONS TRUC TIVO	RUIDO	GRASP	BEST SHIFT	FIRST SHIFT	OTHER LOCALS
13	50.81	50.81	50.81	50.81	55.56	55.56	89.71	50.81	50.81	55.56
14	11.78	11.78	11.78	11.78	11.78	11.78	17.70	11.78	11.78	11.78
15	97.4 6	92.26	97.82	89.59	123.8 6	108.05	115.50	100.9 0	99.91	108.05
16	91.82	84.97	95.55	95.55	133.8 2	124.13	147.8 9	128.34	121.28	147.89
17	157.7 7	150.8 0	155.55	154.16	182.5 3	181.97	171.34	162.12	161.65	171.34
18	144.1 8	138.6 0	139.4 2	139.4 2	171.6 4	170.35	151.92	157.44	150.2 0	170.35

QUÉ SE PUEDE DECIR

En cuanto a la distancia total, el GA obtuvo un GAP promedio de 97.99%, mientras que el GA-VND alcanzó un GAP promedio de 96.16%. Los cuales son similares a los GAPs alcanzados por los algoritmos paralelizados SA y VNS (97.78% y 96.11%, respectivamente).



IL CAMPEONE

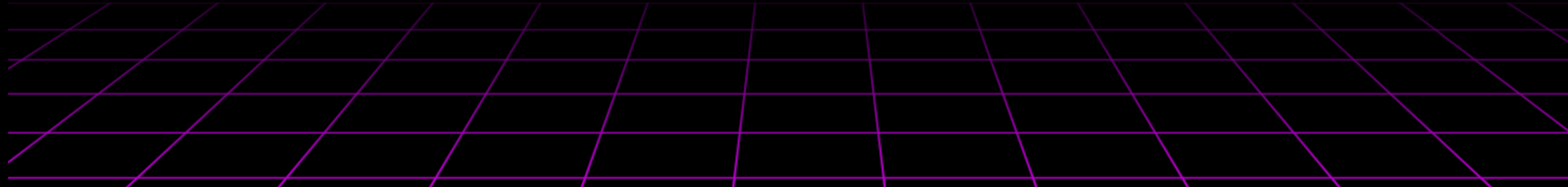
Así, dado que ambos algoritmos genéticos obtuvieron mejores GAP promedio en cuanto a número de vehículos y GAPs promedio comparables en cuanto a distancia, es claro que son los ganadores entre todos los algoritmos desarrollados durante el curso. El GA-VND es claramente el mejor, pues combina el GA original con una última fase de VND para mejorar la solución final.

TIEMPOS DE CÓMPUTO

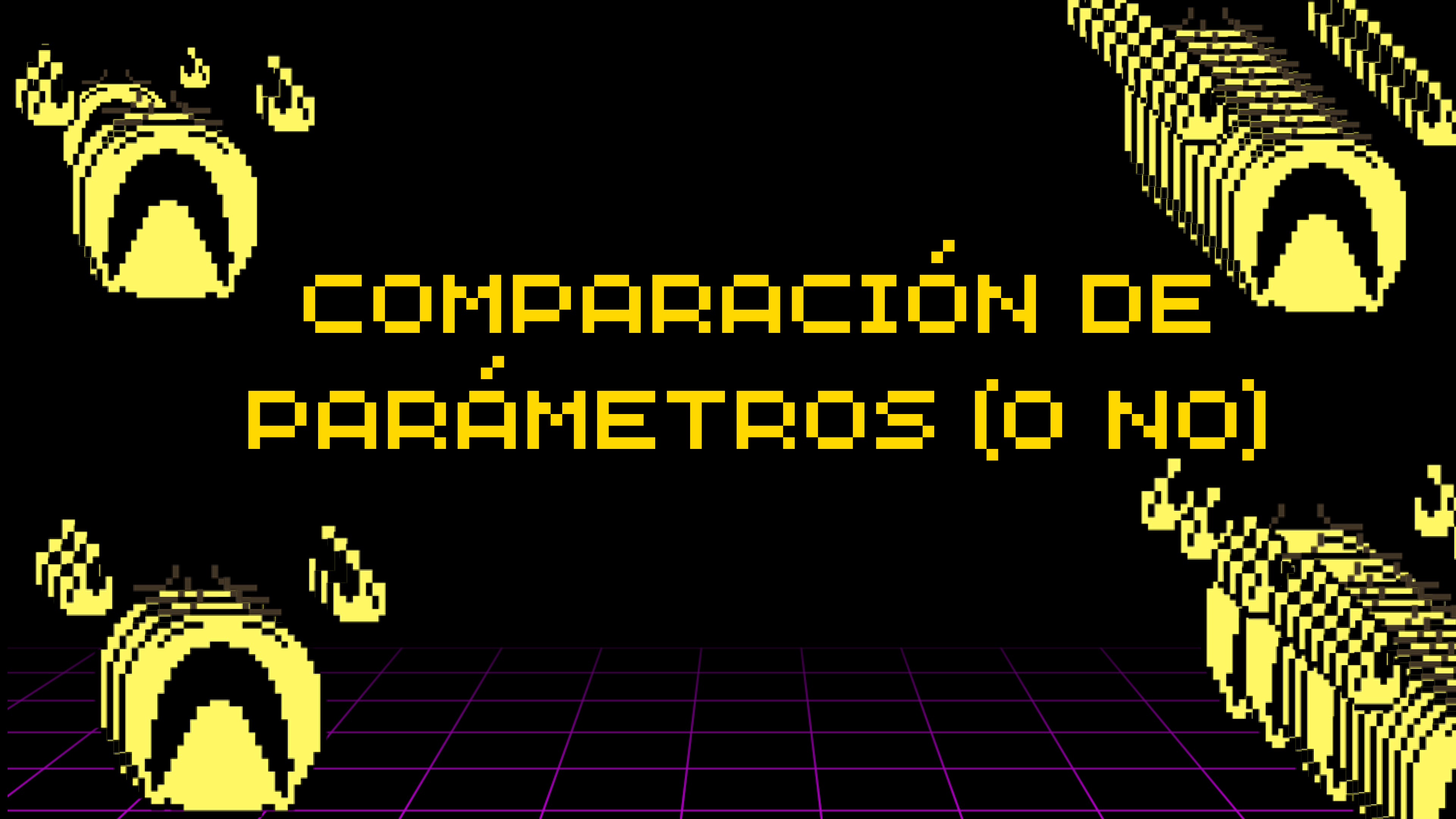
Los tiempos promedios de ejecución en segundos fueron de 0 para el constructivo, 0 para el de ruido y 0 para el GRASP. Para first shift, best shift, first exchange, best exchange, first rearrange y best rearrange fue de 0.18, 0.21, 0.03, 0.03, 0.02 y 0.02 segundos, respectivamente.

TIEMPOS DE CÓMPUTO

Asimismo, para el parallel VNS fue de 281 segundos y de 231 segundos para el parallel simulated annealing. Para el GA fue de 54.51 segundos y para el GA-VND fue de 64.39 segundos, que concuerda con que tienen una última fase adicional.



COMPARACIÓN DE PARÁMETROS (O NO)

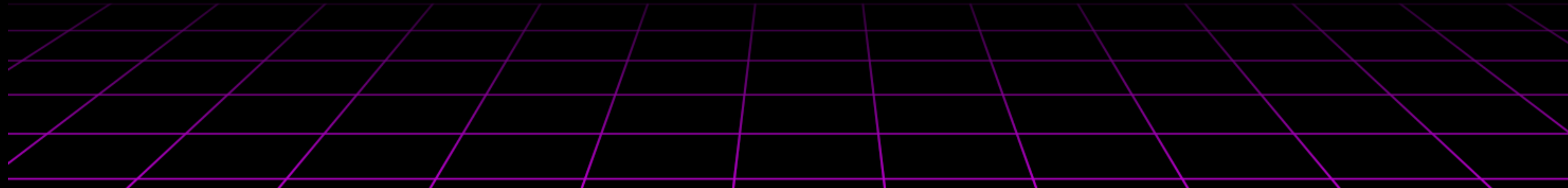


DIFERENCIA

Los métodos resultaron ser relativamente sensible al cambio de parámetros. Sin embargo, si el número de generaciones era ≥ 200 , siempre parecía alcanzar similar punto de equilibrio. Por otro lado, una mayor tasa de crossover o de mutación parecía afectar simplemente la velocidad de convergencia, no las características de la solución final.

DSAFSD

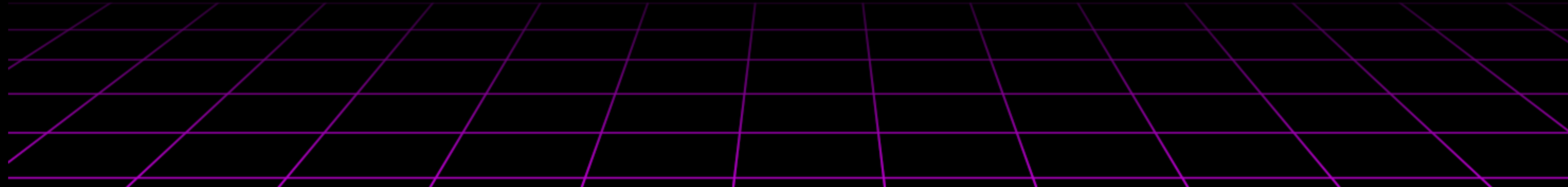
Las soluciones presentadas anteriormente fueron obtenidas con 350 generaciones, 1000 de población, 0.8 crossover rate y 0.1 mutation rate. A continuación se presentan los resultados de la forma
numGenerations_populationSize_crossoverRate_mutationRate.



MÉTODO | GAP % VE | GAP % DIST

VRPTW_CAMILO_GA- 200-250-50-10	102.28%	108.95%
VRPTW_CAMILO_GA- 200-250-80-100	92.22%	101.63%
VRPTW_CAMILO_GA- 50-250-80-100	96.73%	104.35%
VRPTW_CAMILO_GA- 200-250-10-10	105.32%	120.18%

CONCLUSIONES



SORPRESAS TE DA LA VIDA

- Al principio no esperaba que los algoritmos evolutivos superaran (ni siquiera que igualaran) el simulated annealing, pero me vi sorprendido por la facilidad con la que alcanzaban buenas soluciones y asimismo la rapidez con la que lo hacían.
- La última fase de VND logró una mejora pequeña pero significativa en cuanto a la distancia total de las soluciones.
- Valdría la pena paralelizar estos algoritmos genéticos para ver si se obtiene una mejora en los resultados.



GRACIAS!

EL FIN